图文详解
易学实用

从基础操作到高效办公应用

Excel 2010 VBA 入门与实战

刘增杰 王英英 编著

408个示例和3个综合案例
掌握Excel VBA应用技术

清华大学出版社

Excel 2010 VBA 入门与实战

刘增杰 王英英 编著

清华大学出版社
北 京

内 容 简 介

本书内容主要包括 Excel 2010 开发平台概述、Excel 2010 VBA 的开发环境、宏概述、VBA 基础、程序控制结构、使用数组、使用过程、Excel 工作簿相关操作、Excel 工作表相关操作、Excel 单元格相关操作、控制工具与窗体控件的相关操作、窗体控件的实际应用、自定义函数的实际应用、用 Excel VBA 实现数据分析和高效文件处理等。最后通过开发员工信息表、员工薪金管理系统和企业收支管理系统 3 个综合案例，使读者进一步巩固所学的知识，提高综合实战能力。

本书内容全面，实例丰富，图文并茂，步骤清晰，通俗易懂，重点介绍 Excel VBA 在办公实战中的应用技巧，注重实战操作，使读者能够快速入门，循序渐进地掌握 Excel VBA 中的各项技能。另外，作者专门建立了一个 QQ 技术学习群，读者可以和作者面对面交流，解决自学中的困惑，快速掌握 VBA 的相关技能。

本书适合广大 Excel VBA 新手、应用开发人员和办公管理人员学习和参考。

图书在版编目（CIP）数据

Excel 2010 VBA 入门与实战/刘增杰，王英英编著. – 北京：清华大学出版社，2012.9

ISBN978-7-302-29624-9

I. ①E… II. ①刘… ②王… III. ①表处理软件Ⅳ. ①TP391.13

中国版本图书馆 CIP 数据核字（2012）第 179478 号

责任编辑：夏非彼
封面设计：王　翔
责任校对：李金平
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京天颖印刷有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张：**27.5　　**字　　数：**704 千字

版　　次：2012 年 9 月第 1 版　　**印　　次：**2012 年 9 月第 1 次印刷
印　　数：1~4000
定　　价：55.00 元

产品编号：045069-01

前　言

本书是一本面向 Excel VBA 初学者的高质量的书籍。通过详细的实用案例，让读者快速入门，掌握 Excel VBA 应用技巧，提高工作的效率，再也不会为眼前的一堆数据的分析和处理而发愁。

本书特色

内容全面：知识点由浅入深，涵盖了所有 Excel VBA 的基础知识点，由浅入深地介绍了 Excel VBA 开发技术。

图文并茂：注重操作，图文并茂，在介绍案例的过程中，每一个操作均有对应步骤和过程说明。这种结合的方式使读者在学习过程中能够直观、清晰地看到操作的过程及效果，便于更快地理解和掌握。

易学易用：颠覆传统“看”书的观念，变成一本能“操作”的图书。

案例丰富：把知识点融汇于系统的案例实训中，并且结合综合案例进行讲解和拓展，进而达到“知其然并知其所以然”的效果。本书提供了 408 个详细实例和 3 个综合案例源代码，让读者在实战应用中掌握 Excel VBA 的每一项技能。

提示技巧：本书对读者在学习过程中可能会遇到的疑难问题以“提示”和“技巧”的形式进行了说明，以免读者在学习的过程中走弯路。

技术支持：本书作者的技术支持 QQ 群号为 221376441，若读者在本书学习中碰到问题，可以通过 QQ 与作者交流，解决学习中的困惑，快速掌握相关知识点。

读者对象

本书是一本完整介绍 Excel VBA 的教程，内容丰富，条理清晰，实用性强，适合如下读者学习使用：

- 对 Excel VBA 有兴趣，希望快速、全面地掌握 Excel VBA 的读者。
- 对 Excel VBA 完全不了解或者有一定了解，想学习 Excel VBA 并进行应用开发的读者。

鸣谢

本书作者长期从事 Excel VBA 实训工作。参与本书编写人员除了封面署名人员以外，还有苏士辉、肖品、孙若淞、宋冰冰、王维维、梁云亮、陈伟光、刘玉萍、王攀登、张工厂、王婷婷、姬远鹏、臧顺娟和程铖等人。本书虽然倾注了编者的努力，但由于水平有限、时间仓促，书中难免有错漏之处，请读者批评指正，如果遇到问题或有意见，敬请与我们联系，我们将全力提供帮助。

示例下载

本书示例可以在http://download.csdn.net/detail/brucexia/4480264 下载。如果链接失效，可以直接在 QQ 上群里向作者资料。

编　者

2012 年 8 月

目　录



第 1 章　Excel 2010 开发平台概述

在学习 VBA 应用程序开发前，用户需要了解其开发平台 Excel 2010 的相关知识，以有利于后期的开发工作。本章主要讲述初识 Excel 2010、用 Excel 2010 开发应用程序的优势、Excel 应用程序的结构和 Excel 应用程序开发流程等知识。



1.1 初识 Excel 2010

和以往版本相比，Excel 2010 采用了全新的工作界面。启动 Excel 2010 后，即可看到 Excel 2010 的工作界面，如图 1-1 所示。



应聘者信息表

序号	应聘岗位	申请人姓名	性别	年龄	初步简历筛选是否通过	最高学历	面试日期	面试具体时间	邮箱地址	面试地点
1	销售	王	男	21	是	大专	2010.3.30	上午10时30分	123@163.com	某区某路30号某公司人力资源部
2	销售	李	男	22	是	本科	2010.3.31	上午10时30分	123@163.com	某区某路30号某公司人力资源部
3	销售	刘	男	24	是	本科	2010.3.32	上午10时30分	123@163.com	某区某路30号某公司人力资源部
4	销售	段	男	26	是	本科	2010.3.33	上午10时30分	123@163.com	某区某路30号某公司人力资源部
5	销售	郑	男	20	是	本科	2010.3.34	上午10时30分	123@163.com	某区某路30号某公司人力资源部
6	销售	孙	女	25	是	大专	2010.3.35	上午10时30分	123@163.com	某区某路30号某公司人力资源部
7	销售	陈	女	28	是	大专	2010.3.36	上午10时30分	123@163.com	某区某路30号某公司人力资源部
8	销售	杨	女	27	是	大专	2010.3.37	上午10时30分	123@163.com	某区某路30号某公司人力资源部
9	销售	张	女	26	是	大专	2010.3.38	上午10时30分	123@163.com	某区某路30号某公司人力资源部
10	销售	吴	男	23	是	技校	2010.3.39	下午3时30分	123@163.com	某区某路30号某公司人力资源部
11	销售	马	女	21	是	高中	2010.3.40	下午3时30分	123@163.com	某区某路30号某公司人力资源部
12	销售主管	林	男	29	是	研究生	2010.3.41	下午4时30分	123@163.com	某区某路30号某公司人力资源部
13	文员	贺	女	20	是	本科	2010.3.42	下午5时30分	123@163.com	某区某路30号某公司人力资源部
14	经理助理	宋	女	21	是	本科	2010.3.43	下午6时30分	123@163.com	某区某路30号某公司人力资源部
15	工程师	魏	男	35	是	博士	2010.3.44	下午7时30分	123@163.com	某区某路30号某公司人力资源部

图 1-1　Excel 2010 的工作界面

1.1.1 Excel 2010 的任务

Excel 2010 是一个功能强大、技术先进、使用方便的表格式数据综合处理和分析软件，它以电子表格的方式进行数据处理，并可将数据以图形化的方式表现出来，同时还具备简捷、灵活的图形化操作窗口。

具体的任务体现如下。

1. 表格处理

Excel 2010 中所有的数据信息都是以二维表格形式（电子表格）进行存储和管理的，单元格中数据间的关系一目了然，数据的处理更直观、更方便、更易于理解。从形式上看，Excel 2010 与手工表格颇为类似，但 Excel 2010 的电子表格中不仅仅有各种数据，还包括一些计算公式和函数。当在电子表格中输入数据时，这些公式和函数能自动完成所需的计算和分析，大大提高了数据处理的效率。

2. 数据分析

Excel 2010 的数据分析功能提供了财务、日期与时间、数学与三角函数、统计、查找与引用、数据库、文本、逻辑和信息等几百个内置函数，可以满足许多领域内数据处理与分析的要求。如果内置函数不能满足需要，用户还可以使用 Excel 2010 内置的 Visual Basic for Application（VBA）来创建自定义函数。

另外，Excel 2010 除了具有一般数据库软件所提供的数据排序、筛选、查询、统计汇总等数据处理功能外，还提供了相应的数据分析和辅助决策工具，包括数据透视表、模拟运算表、假设检验、方差分析、移动平均、指数平滑、回归分析、规划求解、多方案管理器等分析工具。用户只要正确地选择适当的参数，即可利用以上工具完成复杂的分析求解过程，得到相应的分析结果和完整的求解报告。

3. 图表制作

与同类软件相比，Excel 2010 具有极强的图表处理功能，可以方便地将数据表格中的有关数据转化为专业化图表。此外，它提供了丰富的图表类型，有条形图、柱形图、折线图、散点图以及多种复合图表和三维图表，并针对每一种图表类型提供了几种不同的图表格式，以供用户自动套用。

如果 Excel 2010 所提供的标准图表类型不能满足特殊需要，用户还可自定义图表类型，并可对图表的标题、数值、坐标及图例等项目分别进行定制，从而获得最佳的外观效果。同时，Excel 2010 还能够自动建立数据与图表间的关联，当数据更新时，图表随数据的变化而变化。

4. 宏和 VBA 功能

为了更好地发挥 Excel 2010 的强大功能，Excel 2010 还提供宏功能以及内置的 VBA，用户可以使用它们来进行二次开发，拓展和提高 Excel 的应用水平。

宏是为初级用户提高工作效率而提供的。Excel 2010 提供有宏记录器，可将用户所做的操作记录下来，自动转换成相应的宏命令。当需要再次执行这些操作时，用户可直接运行这些宏命令。

对于需要经常使用的宏，可将其指派给特定的自定义菜单或自定义工具按钮，以后只要选择相应的菜单命令或单击相应的工具按钮，就可以直接运行这些宏了。

利用 Excel 2010 提供的 VBA 功能，可以充分发挥 Excel 的各个功能，实现 Excel 软件的更高层次应用（如程序编制、软件开发）。

1.1.2　Excel 2010 的新增功能

Excel 2010 与以前的版本相比，增加了许多新功能，为将来实现人性化办公提供了条件。

1. 更直观地表达想法

Excel 2010 开创了一些设计方法，让用户可以将想法生动地表达出来。使用新增的和改进的图片格式工具（例如，颜色饱和度和艺术效果）可以将文档画面转换为艺术品。在 Excel 2010 中，将这些工具与大量预置的新主题和 SmartArt 图形布局配合使用，可以更淋漓尽致地表达出自己的想法，如图 1-2 所示。

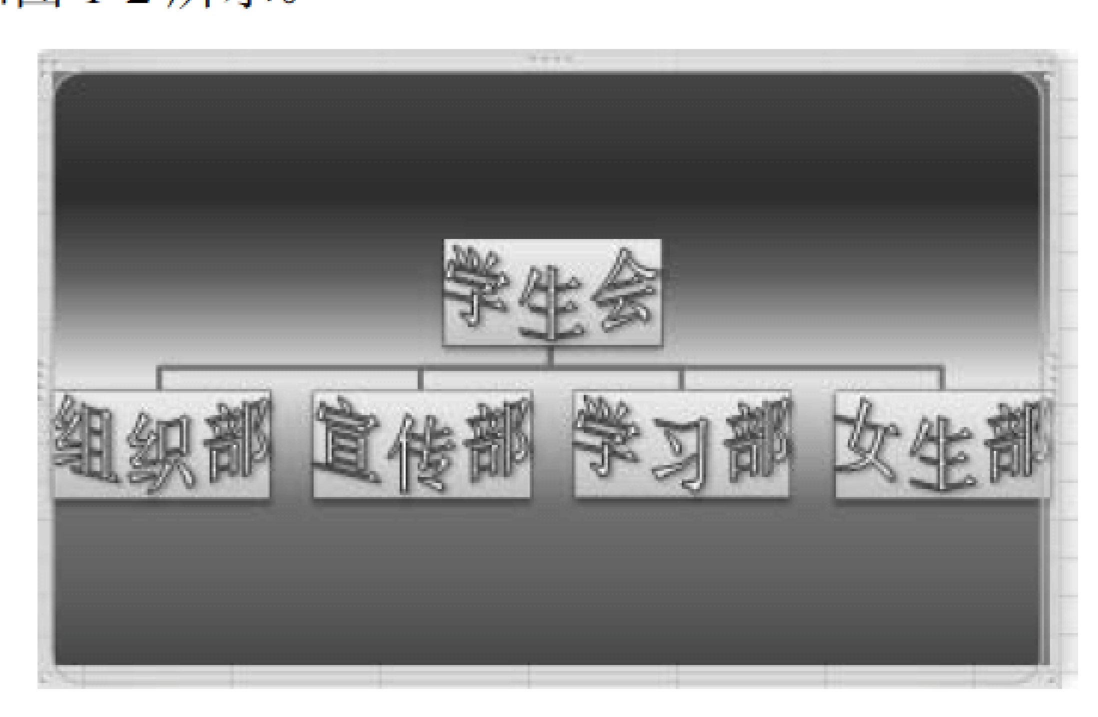


图 1-2　SmartArt 布局的图形

2. 更加人性化的功能区

利用功能区，用户可以轻松地查找以前隐藏在复杂菜单和工具栏中的命令和功能。除此之外，Excel 2010 添加了更加强大的自定义功能区的功能，用户可以创建自己的选项卡和组，还可以重命名或更改内置选项卡和组的顺序。如图 1-3 所示为自定义的功能区。

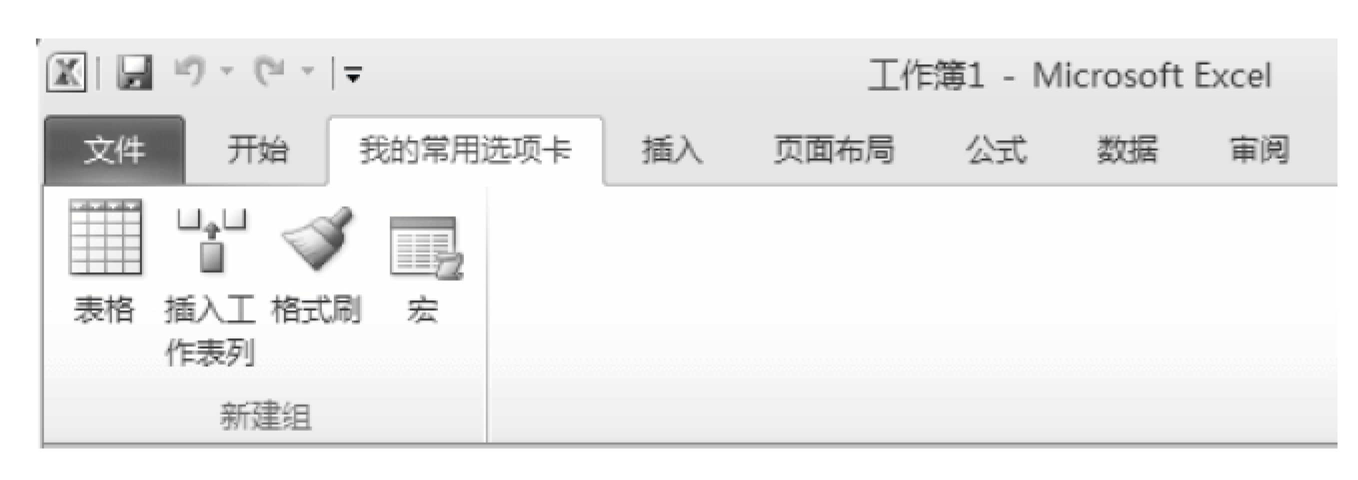


图 1-3　自定义的功能区



3. 从更多地点、更多设备上享受熟悉的 Excel 体验

使用 Excel 2010，可以从更多地点、更多设备上更轻松地完成任务。可以在智能手机或几乎每台连接至 Internet 的计算机上，随时随地进行工作。

Microsoft Office Web Apps 将 Excel 2010 体验扩展到 Web，可以在线存储 Excel 文件，然后通过 Web 访问、查看、编辑和共享文件内容，如图 1-4 所示。



图 1-4　保存到 Web 功能

4. 提供强大的数据分析和可视化功能

使用 Excel 2010 中的数据分析和可视化功能可以跟踪和亮显重要的趋势，如图 1-5 所示。使用新增的 Sparklines 功能可以在工作表单元格中使用小图表来清晰简洁地表达数据。使用 Slicers 功能可以在多个层对 PivotTable 数据进行过滤和拆分，从而可以减少格式设置时间，增加分析时间。

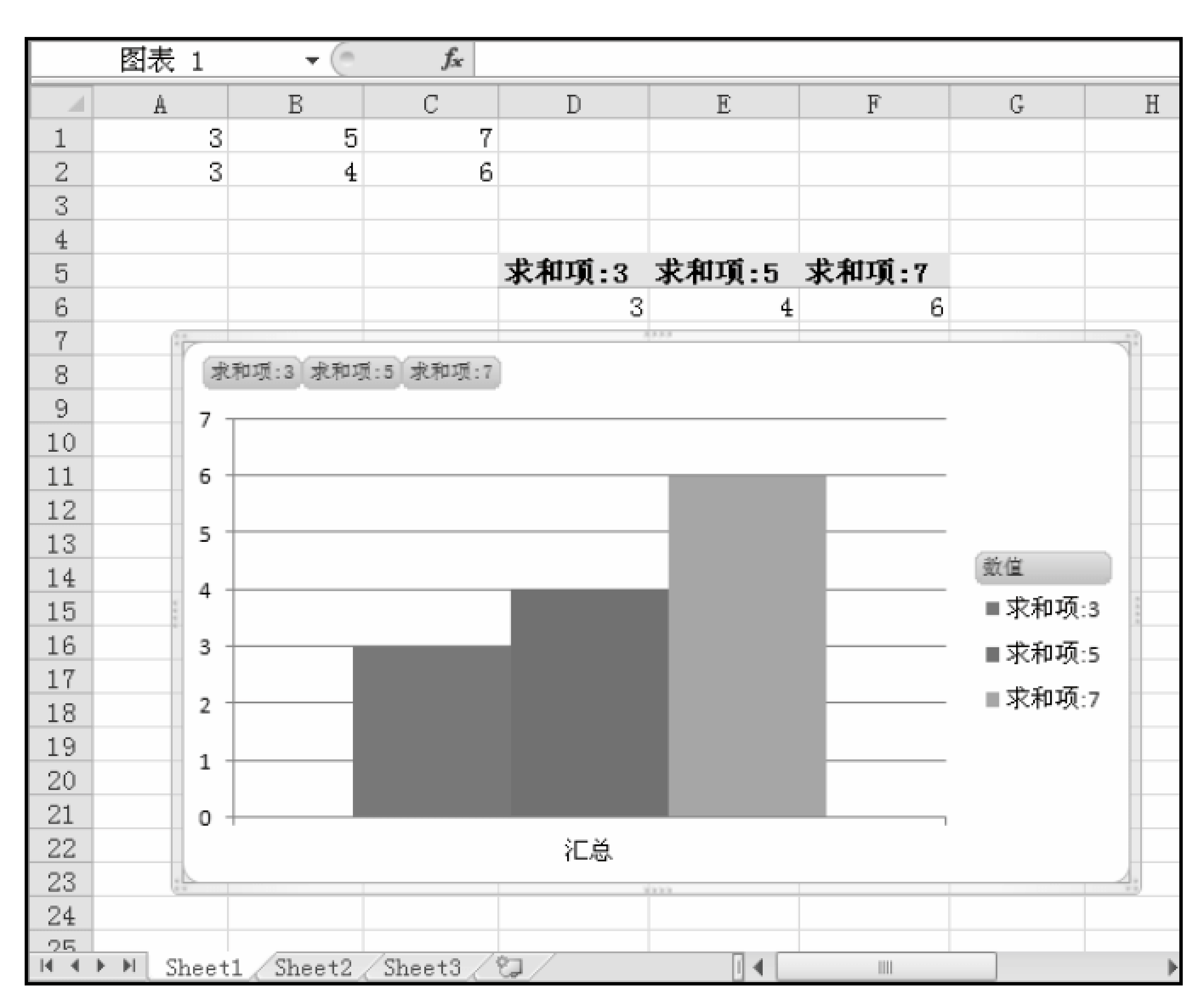


图 1-5　强大的数据分析和可视化功能

5. Backstage 视图

Backstage 视图是 Microsoft Excel 2010 程序中的新增功能，它是 Microsoft Office Fluent 用户界面的最新创新技术，并且是功能区的配套功能。选择【文件】菜单即可打开 Backstage 视图，如图 1-6 所示，用户可在此打开、保存、打印、共享和管理文件以及设置程序选项。



图 1-6　Backstage 视图

6. 快速、有效地比较数据列表

在 Excel 2010 中，迷你图和切片器等新增功能以及对数据透视表和其他现有功能的改进可帮助用户了解数据中的模式或趋势，从而做出更明智的决策，如图 1-7 所示。

政治经济学	电子商务概论	计算机基础	管理学	总分	综合测评分
80	77	80	90	327	213.6
90	78	81	91	340	218
60	79	82	92	313	214.4
77	80	83	93	333	220.2
86	81	84	94	345	224.4
78	82	85	95	340	225.2
92	83	86	96	357	230.4
85	84	87	97	353	231.4
78	85	88	98	349	232.4
99	86	89	99	373	239

图 1-7　迷你图

1.1.3　Excel 2010 的界面

Excel 的工作界面主要包括标题栏、功能区、工作表编辑区、状态栏等部分，如图 1-8 所

示。通过该工作界面，用户可以更快速、更轻松地找到各种工具，从而提高工作的效率。

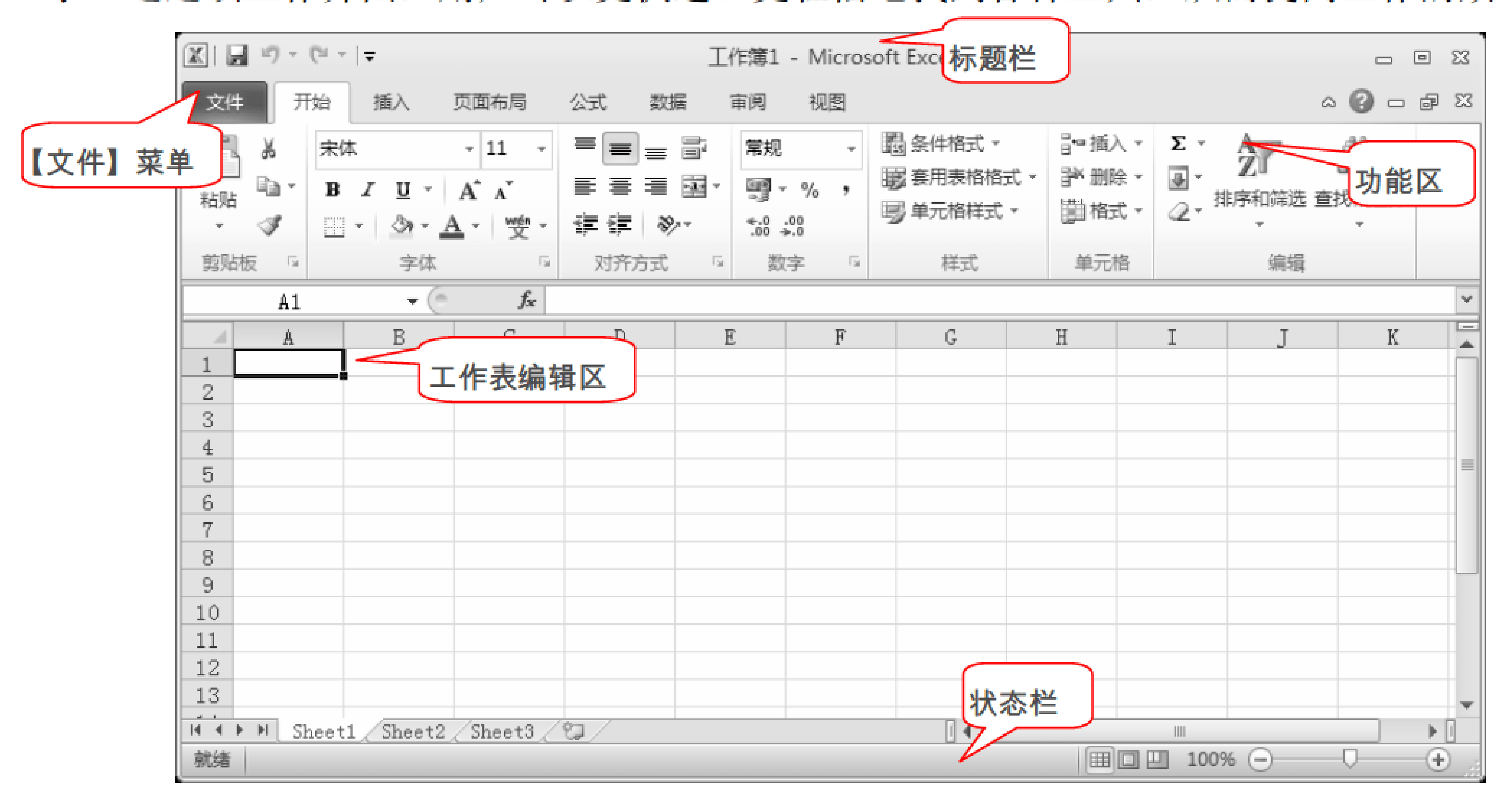


图 1-8　Excel 2010 的工作界面

1. 【文件】菜单

【文件】菜单位于 Excel 2010 工作界面的左上角，用绿色显示，选择【文件】菜单，在弹出的下拉菜单中不但包含有 Excel 早期版本所具有的【打开】、【保存】和【打印】等命令，还新增了【信息】等命令。另外，在下拉菜单的右侧显示了最近使用过的文档，如图 1-9 所示。



图 1-9　【文件】菜单

2. 标题栏

Excel 2010 的标题栏也与以前的版本有所不同。默认状态下，【标题栏】左侧显示的是【快速访问工具栏】，中间显示的是当前表格的文件名称，右侧显示的是窗口控制按钮。其中，窗口控制按钮包括【最小化】、【最大化】和【关闭】等 3 个按钮。启动 Excel 2010 后，标题栏中显示的默认文件名为“工作簿 1”。

3. 快速访问工具栏

Excel 2010 的快速访问工具栏中包含最常用操作的快捷按钮，方便用户使用。在默认状态中，快速访问工具栏中包含【保存】、【撤销】和【恢复】按钮，如图 1-10 所示。

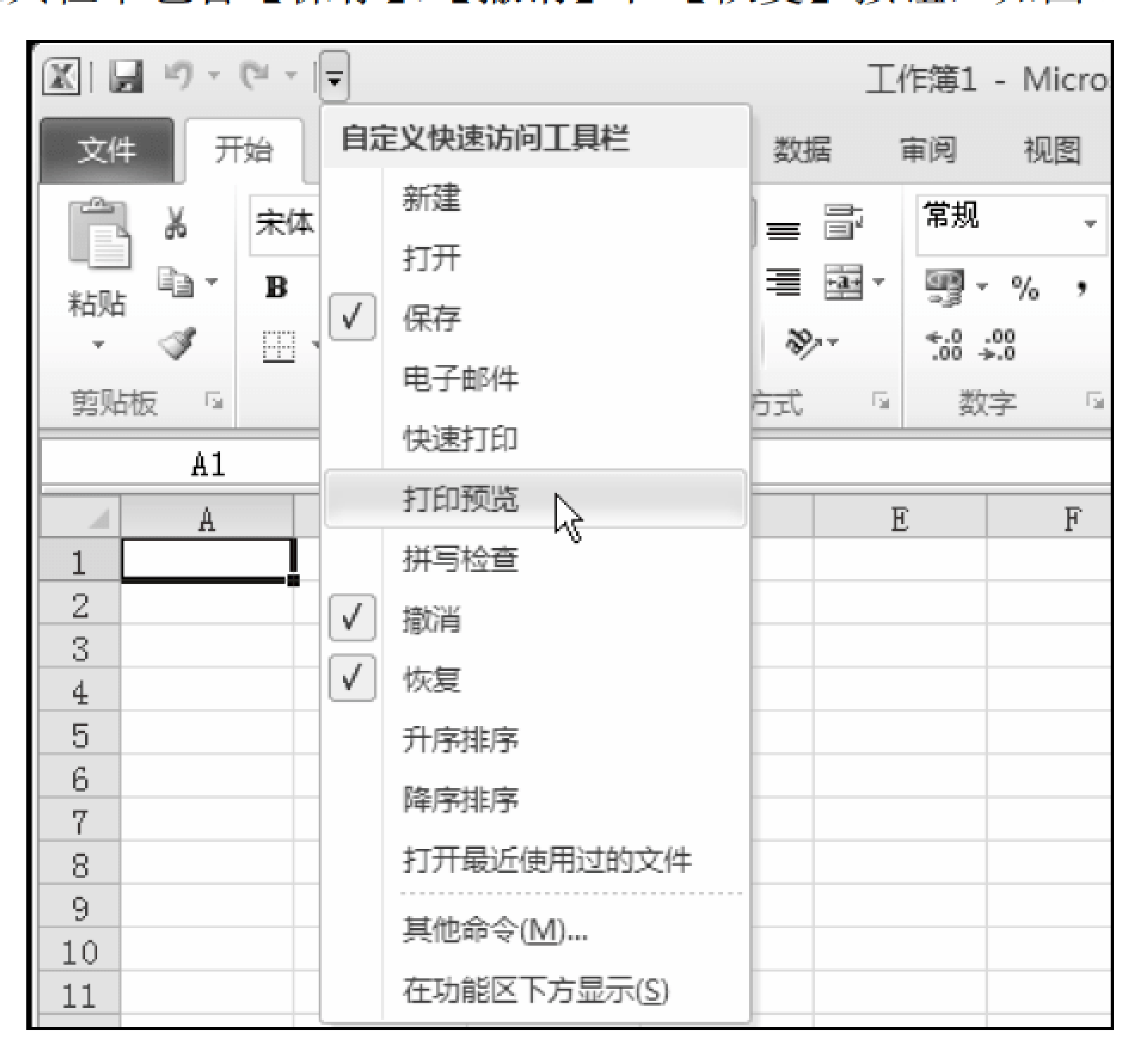


图 1-10　快速访问工具栏

如果想在快速访问工具栏中添加或删除其他按钮，可以单击【自定义快速访问工具栏】按钮，在弹出的下拉列表中选择需要在快速访问工具栏中显示的命令。在该下拉列表中，可以看到被显示的命令前面有一个对号表示处于被选中的状态。

4. 功能区选项卡

功能区选项卡是一种特殊的选项板，位于工作表编辑区的上方，是菜单栏和工具栏的主要替代工具。默认状态下，功能区选项卡中包含【开始】、【插入】、【页面布局】、【公式】、【数据】、【审阅】、【视图】等选项卡，每个选项卡包含若干个选项组，每个选项组又包含许多由图标表示的命令按钮，如图 1-11 所示。为了方便运用，某些选项卡只有在需要时才显示。



图 1-11　功能区选项卡

另外，在功能区选项卡的标签最右侧显示了工作簿的一些控制按钮，包括【帮助】、【最小化】、【最大化】和【关闭】等 4 个按钮。单击【帮助】按钮或直接按【F1】键，可以打开【Excel 帮助】窗口，如图 1-12 所示。

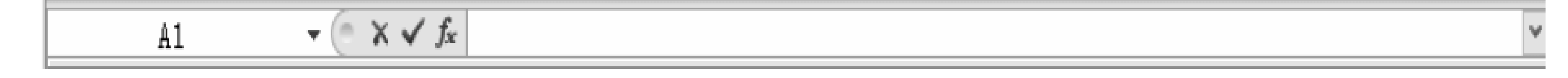


图 1-12　【Excel 帮助】窗口

5. 编辑栏

编辑栏位于功能区的下方、工作表编辑区的上方。编辑栏用于显示和编辑当前活动单元格中的数据和公式。

编辑栏由 3 部分组成，分别为左侧的名称框、中间的功能按钮和右侧的公式框。



- 名称框：用于显示当前被选中单元格的地址和名称。

- 功能按钮：当向单元格中输入数据时，位于名称框和公式框之间的功能按钮显示，包括【确定】按钮✓和【取消】按钮✗。单击【确定】按钮✓，可以确定该单元格的内容，退出编辑状态；单击【取消】按钮✗，可以取消对该单元格的编辑。
- 公式框：主要用于向活动单元格中输入、修改数据。单击公式框左侧的【插入函数】按钮fx，弹出【插入函数】对话框，如图 1-13 所示。可以通过该对话框查找和插入需要的函数。



图 1-13　【插入函数】对话框

6. 工作表编辑区

工作表编辑区是 Excel 2010 操作界面中用于输入数据的区域，由位于左侧的行号、下侧的列号、左下侧的工作表标签、右侧与下侧的滚动条以及中间的单元格组成，如图 1-14 所示。在单元格中可以输入不同类型的数据。

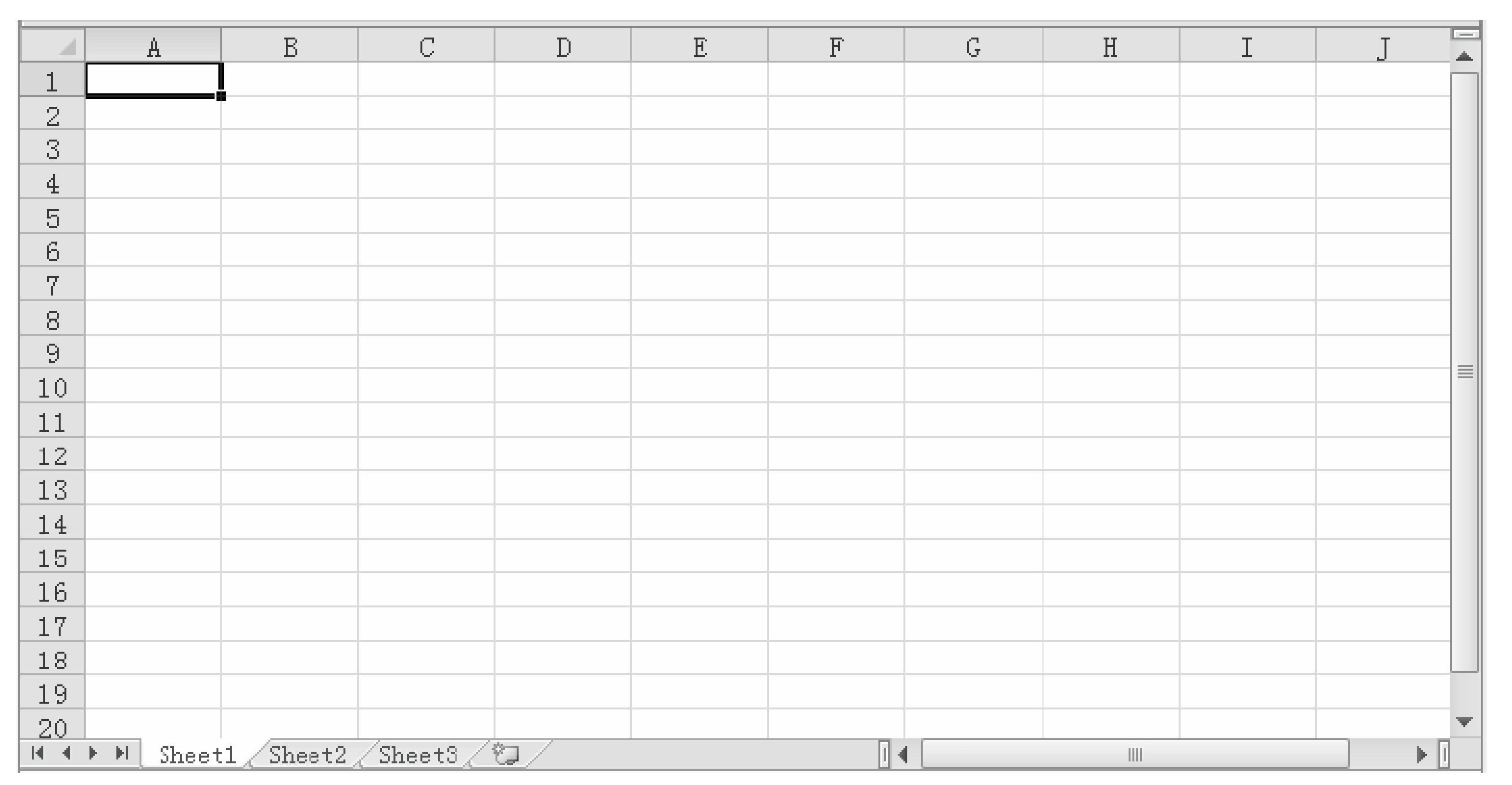


图 1-14　工作表编辑区

7. 状态栏

状态栏位于工作界面的最下方，用于显示当前数据的编辑情况和调整页面显示比例，如图 1-15 所示。



图 1-15　状态栏

页面显示控制区包括常用视图按钮和页面显示比例大小控制滑块。

页面显示控制区从左到右依次是【普通】、【页面布局】和【分页预览】视图按钮，以及页面显示比例大小控制滑块。以黄色为底表示当前正在使用的视图方式；缩放级别随着滑块的拖动而改变，向左拖动滑块减小文档显示比例，向右拖动滑块增大文档显示比例。



1.2 用 Excel 2010 开发应用程序的优势

与一般的开发平台相比，使用 Excel VBA 开发应用程序有自己独特的优势。

（1）超快的开发速度。一般情况下，客户对产品的需求比较着急，采用其他语言开发，周期长，而客户最关注的是开发的应用程序能否满足工作的需求，而不是用哪种语音开发。使用 Excel VBA 开发应用程序的周期一般是其他语言的一半，甚至更短，提高了项目完成的效率。

（2）功能强大易开发。用户可以直接在 Excel 中创建 VBA 应用程序，与其他语言相比更容易学习和使用，而且功能十分强大，能实现很多复杂的功能。

（3）代码重复利用率高：当代码编写完成后，只要编码规范，有很好的注释，二次开发时工作比较轻松，代码可以重复利用。

（4）界面设计效率高。和其他语言相比，使用 Excel VBA 制作用户界面的速度很快，非常容易添加控件，例如按钮、复选框和列表框等，实现这些控件的功能几乎不需要任何宏代码。

（5）定制各个界面元素。用户可以使用窗体很容易创造各个界面元素，包括十分专业的对话框，右击弹出的各种快捷菜单、工具栏和菜单栏。

（6）自定义工作表函数。使用 VBA 可以创建各种需求的函数公式，完成复杂的数据处理工作。

（7）强大的数据分析功能。使用 Excel 2010 的数据透视表功能可以很容易地对大量的数据进行汇总，从而大大提高工作效率。

（8）访问外部数据很容易。通过提供 ADO 和 DAO 两个对象模型，使 VBA 应用程序很容易访问外部数据。



1.3 Excel 应用程序结构

使用 VBA 创建 Excel 应用程序有一定的结构规范，下面将简单介绍其构成。

1.3.1 Excel 应用程序的构成

从开发者角度分析，Excel VBA 应用程序由工作表、用户窗体、模块和类模块等部分组成。

工作表主要对数据进行组织和分析，是程序的主体部分。在工作表中设置好样式和格式，通过 VBA 代码获取表格中的数据，经过加工处理后将其显示在相应的单元格中，以便于用户进行查看或打印等操作。

在 Excel VBA 应用程序中，用户除了可以使用工作表直接进行交互外，还可以添加窗体，用来和用户进行操作。使用窗体的好处是：可以将用户与数据表中的数据很好地隔离，提高了数据的安全性，使不同权限的用户看到不同的数据范围。

模块中可以保存程序的通用过程，其他程序可以直接调用。在 Excel VBA 中用户可以添加、移除、导入和导出模块等操作。

在 Excel VBA 应用程序开发的过程中，用户既可以使用系统自带的对象，也可以自定义对象，只要将自定义的类保存在类模块中即可。

1.3.2 面向对象编程机制

和大部分语言编程一样，Excel VBA 也支持面向对象编程机制。对象是对现实世界实体的模拟，由实现实体的过程来定义。一个对象可被认为是一个把数据和事件封装在一起的实体，这个程序产生该对象的动作。事件可以通过单击鼠标或者按键的操作产生，也可以通过一个窗口的操作产生等。在 VBA 中，通过事件驱动提供开发人员与系统的接口，开发人员通过编写事件过程来处理该事件时需要完成的工作，例如单击鼠标可以关闭一个对话框。

在面向对象编程机制中，代码的执行顺序不按照预定的顺序执行，而是根据不同的事件执行不同的代码。事件的触发方式很多，既可以是用户直接操作触发，也可以是操作系统或其他应用程序的消息触发，还可以是应用程序本身的消息触发。所以事件的顺序决定了代码执行的顺序，每次代码运行的路径会有所不同。VBA 编写开发应用程序，实际上是编写不同对象的事件代码。



1.4 Excel 应用程序开发流程

了解应用程序的开发流程，对提高工作效率有很大的帮助。在进行 Excel VBA 应用程序开发的过程中，读者可以参照 Visual Basic 开发应用程序的方法，总结 Excel 应用程序开发的一般流程。

1.4.1 需求分析

在进行 Excel 应用程序开发前，需要仔细分析用户的需求。需求分析如果做不好，会直接影响后续的开发工作。一旦需求分析出现大的失误，程序开发工作将会从头再来，严重影响工作效率。

所谓需求分析，是指对要开发的程序进行详细的分析，弄清应用程序的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。通俗地说，需求分析就是确定计算机要做什么工作。

需求分析的主要任务如下。

（1）功能需求：列举出所开发软件在功能上应做什么，这是最主要的需求。

（2）环境需求：主要是分析软件系统所处的环境需求。具体包括硬件要求、系统要求、Excel 开发平台的版本等。

（3）可靠性需求：各种软件运行时，失效的影响各不相同。在需求分析时，要分析故障出现的概率，按不同的环境运行提出要求。

（4）安全保密工作需求：对客户的保密工作在本阶段进行分析。

（5）用户界面需求：软件与用户界面的友好性是用户能够方便有效地使用软件的关键之一。无论是从用户的使用角度还是从市场角度分析，具有友好用户界面的软件都具有很强的竞争力。在需求分析阶段，要提出用户界面需求分析。

1.4.2 界面设计

界面是用户与应用程序直接交流的桥梁，所以好的应用程序必须有好的人性化界面。开发人员在设计界面时，一定要牢牢把握方便用户操作这一基本原则，并贯穿到设计界面中。

Excel 2010 取消了工具栏和菜单栏，与之代替的是功能面板。在 Excel 2010 中进行界面设计的方式主要有以下几种。

1. 在比较简单的应用程序中

可向工作表中添加按钮或其他控件，然后与宏代码进行绑定。如图 1-16 所示是在工作表中制作的一个员工管理表。

2. 用户窗体

用户界面是应用程序的一个重要组成部分。在 Excel 2010 应用程序中，用户窗体作为应用程序的用户界面部分，将用户的操作和 Excel 工作表中的数据完全隔离开。如图 1-17 所示为设计好的用户窗体。



图 1-16　员工管理系统程序界面

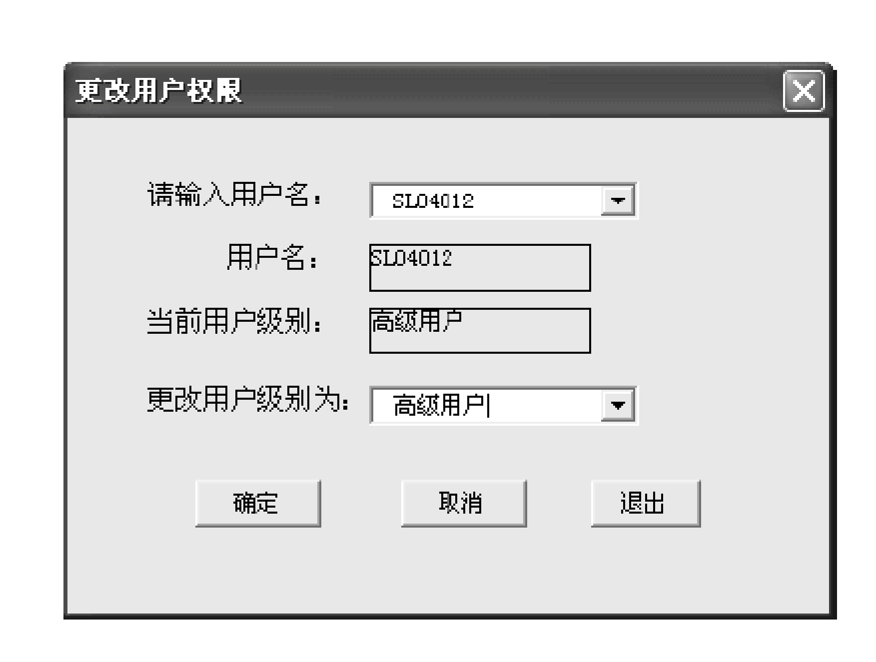


图 1-17　用户窗体

3. 自定义功能区

功能区是 Excel 2010 和早期 2003 版本不同的地方，取代了菜单栏和工具栏。可使用 XML 代码自定义功能区。如图 1-18 所示为自定义的【图书管理系统】功能区。



图 1-18　自定义的【图书管理系统】功能区

1.4.3　设计代码

用户界面设计完成后，要想实现功能操作，就需要添加相关的事件代码，如用户窗体中的代码、功能区中的按钮等。

在 Excel 中设计 VBA 应用程序时，界面设计和代码设计的先后顺序没有严格的规定。用户既可以先设计好界面后编写相关代码，也可以编写好代码后设计相关用户界面元素，并进行绑定操作，这要根据实际的操作来定先后的顺序。

1.4.4　帮助系统

一个完整的 Windows 应用系统往往包括在线电子文档的帮助系统，可以让用户尽快理解和熟悉系统的功能。在 Excel 中设计 VBA 应用程序时，特别是对于大型的应用系统，也可以制作这种帮助系统。如图 1-19 所示为 Word 的在线帮助系统。



图 1-19　Word 的在线帮助系统

1.4.5　测试与调试系统

应用程序设计完成后，用户还必须进行系统的测试和调试工作，这是非常重要的一步。在设计测试的数据时，尽可能多地考虑不同的情况，用不同类型的数据去测试系统的显示结果。在设计测试数据的同时，应编写出测试数据的理想结果，并与应用程序的实际结果进行对比，如果不同，则要做相关的代码调试工作。

如图 1-20 所示为系统数据测试显示的提示对话框。



图 1-20　数据测试显示的提示对话框

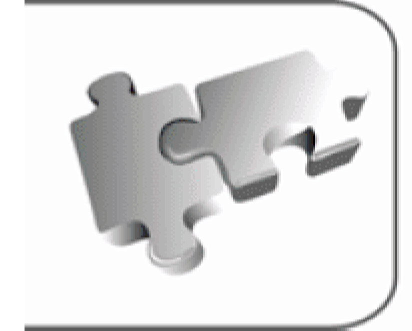
1.4.6　发布系统

系统测试无误后，即可发布给最终用户使用了。在发布时需要考虑以下几个问题。

（1）用户使用环境，如果是在 Excel 2010 环境下开发的应用程序，并使用了 Excel 2010 的新功能，就要求用户也要使用 Excel 2010 版本。如果用户用的是早期版本，则需要将新功能的代码作相关的调整，以适应早期版本的环境需求。

（2）动态链接库。如果应用程序中使用了 ActiveX 控件，则需要考虑是否需要将包含该控件的动态链接库文件也包含在应用程序中一起发布。

（3）相关辅助文件。在一个复杂的应用程序中，可能会包含很多复杂文件，如图片文件、帮助文件、数据库文件等，需要将这些文件也包含在发布文件中。并且最好将其路径修改为相对路径，这样可以在任意盘符下都可以正常访问。



1.5 如何掌握 Excel VBA 编程

要学好 Excel VBA，就要做到如下几点。

（1）保持良好的心态，思路要清晰，扎扎实实，切忌急于求成。

看到 Excel 的一个个新奇的功能，感到很兴奋，于是巴不得一下子全学会，公式与函数、VBA 编程，一股脑儿全开始试验，结果没有明确的目标，觉得很乱，虽然花了很多时间，但是没多大收获。

（2）基本的东西是很重要的，许多复杂的东西都是由简单基本的东西组成的。Excel VBA 也不例外。

对 Excel 的基本命令、操作和功能的了解是关键。大多数人可能有这样的体会，有时候想得很复杂的东西，其实在 Excel 中几个基本的功能就解决了，所以与其一开始就是函数、公式、VBA 什么的，倒不如先将 Excel 的基本功能理解清楚。一般用 Excel 里已存在的功能是最快捷的，要处理的是那些 Excel 的基本功能不能实现的，或者是需要综合运用 Excel 的基本功能来提高工作效率的。同样的道理，只有对基本功能了解清楚了，才能更好地为自己的工作便利而开发利用，从而上升到更高的层次。

还有，就是不要被复杂的东西所吓倒，例如，有一些结果的实现用到了很长、很复杂的公式和函数，其实你把这个公式一分解就知道它是按步骤由简单的公式和函数组合而成的。

再有一点就是，考虑问题要尽可能简单，实用的东西不一定是很复杂的东西，有一些实现结果的方式和程序实际上很简单，也很有用。

（3）要选择好的参考书并经常在优秀的网上论坛交流。

近几年来，特别是 2003 年以来，关于 Office 方面的书越来越多，但大多是讲基本操作的，很简单、重复的内容也很多，此外，还有一些关于 Office 操作技巧方面的书，都不是很深入，这方面的书，看得眼花缭乱，收效也不大，所以选择好的参考书是很重要的一个方面。

在学习过程中还有很重要的一点，就是要经常进行交流。现在网络给我们提供了一个很好的交流平台，在网络上有一些很好的 Excel 网站论坛，可以在上面交流自己的学习成果、学习别人好的做法和经验。同时，对别人提出的问题进行思考和解答，也很有利于自己水平的提高，一方面可以帮助别人解决问题，另一方面也可以看看别的网友是怎样解决问题的，从中获得思路和灵感。但切忌沉迷其中并不深入思考，因为网站论坛上 Excel 的学习资料是很多的，功能技巧和源代码也很多，但这只是便于辅助学习和积累。千万不要沉迷于 Excel 论坛，巴不得把上面的资料和程序全下载到自己的计算机上，下载后资料是很多，但很少去看去思考，还有看到别人实现了某种算法和功能后感到很新奇，自己也模仿但不明白为什么，花费了大量的时间也没学到什么。

（4）多使用，多练习；多思考，多归纳；多记忆，注意学习的方法和技巧，有意识地养

成良好的 Excel 应用操作和 VBA 编程习惯。

基本的东西掌握后，那就是要多使用、多练习了，在使用过程中熟练应用并发现和学习更多新的东西。对发现的问题尽量自己思考解决，对知识点进行系统归纳，一方面便于查找和应用，另一方面也便于记忆。

此外，要注意学习的方法和技巧。其实在学习过程中，遇到问题后，如果参考书上查询较难的话，Excel 的帮助是一个不错的选择，它提供了大量最具价值的参考资料，尤其是在使用 Excel 的公式与函数和 VBA 的时候。

Excel 中的 VBA 也许是大家很感兴趣的，它可以更大范围地发挥 Excel 的强大潜能，但 VBA 很难学，特别是对于没有基础的人来说。可是学好 VBA 是很有好处的，将 Excel 的 VBA 学好了，其他软件上的 VBA 也就不难了，再说 VBA 可以发挥 Excel 更强的功能。刚开始的时候，VBA 是很难掌握的，一旦入门后就顺利了。可以利用 Excel 中的录制宏学习，特别是在编写 VBA 程序的时候，遇到不明白的问题后利用录制宏的功能查看相应功能的实现代码是很有用处的；确定一个开发目标，不断地实践和试验；加强 VBA 源代码的学习和分析。

（5）坚持学习，保持兴趣。

其实学习计算机有一个很大的特点就是，成果可以体现出来，至少可以当即在自己的计算机上实现，有成就感；可以在网站论坛交流，帮助别人解答问题，自己有疑惑也可以提出来，也看看别人所实现的成果，这也可以保持自己的兴趣。此外，现在办公一般都有 Excel，帮助同事和朋友解决问题也是一大乐事，也可以提高自己学习的兴趣和激情。还有，要结合自己的工作实践和解决实际应用问题来学习，也能极大地促进自己的学习兴趣。

第 2 章　Excel 2010 VBA 的开发环境

掌握和运行任何程序之前，熟悉程序的开发环境都是一件很有必要的事情。Excel 提供有 VBE 的开发界面，即 Visual Basic 编辑器（VBE）窗口。在该窗口中用户可以实现应用程序的编写、调试和运行等操作。本章主要讲述 VBE 的简介、VBE 的子窗口、定制 VBE 环境和使用帮助等知识。



2.1 VBE 简介

任何一门语言都有自己的开发环境，VBA 虽然与 Microsoft Office 套装办公软件绑定，但是它也拥有自己的开发界面。Office 提供了 VBE 开发环境，在该窗口中用户可以实现应用程序的编写。但是在使用 VBE 前，必须先运行 Excel 2010 软件。

2.1.1 VBE 概述

所谓 VBE，其实就是 VBA 的编辑环境，即编辑 VBA 代码的工具。在 VBE 开发环境中，用户可以编辑 VBA、调试宏、创建用户窗体、查看或修改对象属性等操作，如图 2-1 所示。

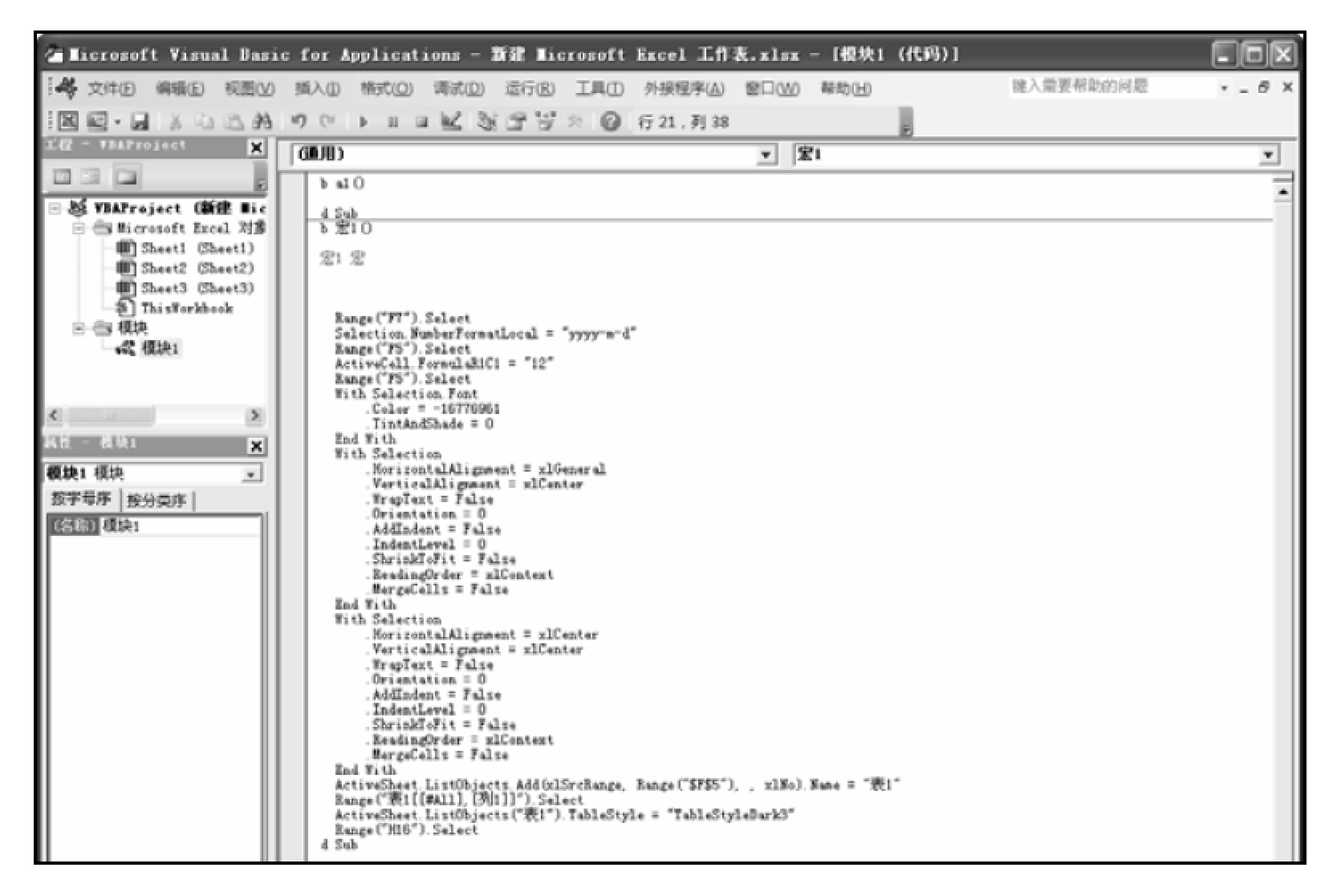


图 2-1　VBE 编辑器窗口

2.1.2 进入 VBE

VBA 编辑器窗口不能单独打开，必须依附于它所支持的应用程序，也就是只有在运行 Excel 的前提下才能打开。

打开 VBA 编辑器窗口有如下几种方法。

01 在 Excel 工作簿中，切换到【开发工具】选项卡中，单击【代码】组中的 Visual Basic 按钮，即可打开 Microsoft Visual Basic for Applications 窗口，如图 2-2 所示。



图 2-2　单击 visual Basic 按钮

02 按 Alt+F11 组合键，也可快速打开 Microsoft Visual Basic for Applications 窗口。

03 如果在工作表中添加了宏，则可在设计模式下选择【视图】➤【宏】➤【查看宏】菜单命令，即可打开 Microsoft Visual Basic for Applications 窗口，如图 2-3 所示。



图 2-3　选择【查看宏】菜单命令

04 在 Excel 工作表控件上右击，在弹出的快捷菜单中选择【查看代码】菜单命令，即可打开 Microsoft Visual Basic for Applications 窗口，如图 2-4 所示。



图 2-4　选择【查看代码】菜单命令

05 在 Excel 工作簿中，切换到【开发工具】选项卡中，单击【控件】组中的【查看代码】按钮，即可打开 Microsoft Visual Basic for Applications 窗口，如图 2-5 所示。



图 2-5　单击【查看代码】按钮

2.1.3　VBE 操作界面

启动 VBE 操作界面以后，即可看到窗口由以下几个部分组成，包括菜单栏、工具栏、工程窗口、属性窗口、代码窗口、用户窗体、立即窗口和本地窗口，如图 2-6 所示。

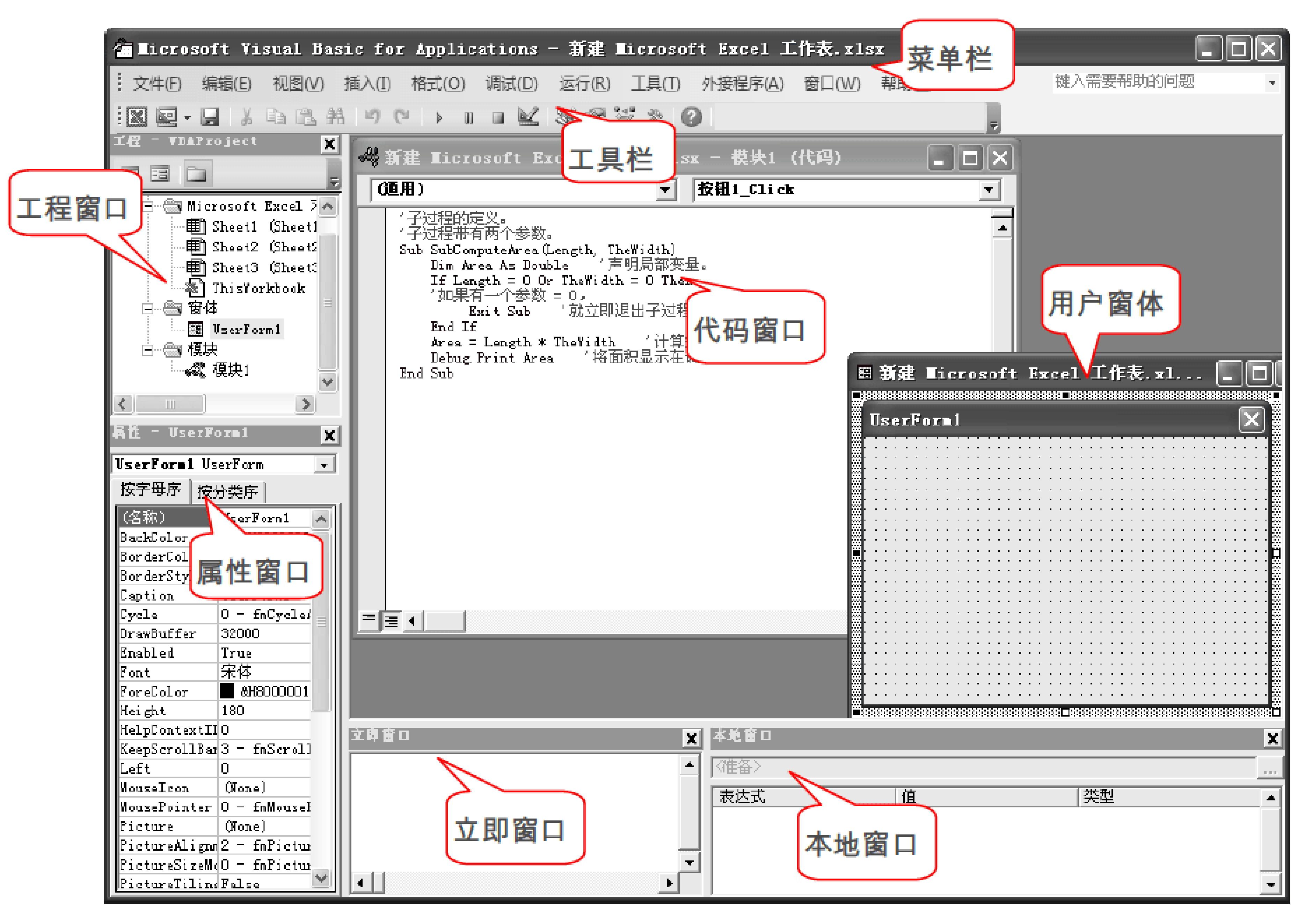


图 2-6　VBE 编辑器窗口

- 菜单栏：位于标题栏的下方，包括【文件】、【编辑】等菜单命令，每个菜单命令都包含若干个菜单命令，分别选择菜单命令中的相关命令就可以执行相应的操作。
- 工具栏：VBA 代码编辑窗口提供有【标准】、【编辑】、【调试】和【用户窗体】4 种工具栏，默认情况下，显示的是【标准】工具栏，用户可以单击菜单栏或工具栏的空白处，在弹出的快捷菜单中选择【编辑】、【调试】或【用户窗体】选项，即可打开相应的工具栏。
- 工程窗口：以树形图示的形式显示了 Excel 工作簿和工作表等对象，其中也有插入的窗体和编制的模块。可以把每一个打开的 Excel 工作簿都看作一个工程，且工程的默认名称为 VBAProject（工作簿名称）。
- 属性窗口：类似于产品说明书，其中列出了所选 Excel 对象的属性及其当前设置。当选定多个



控件时，属性窗口则包含全部已选定控件的属性设置，可以分别切换到【按字母序】或【按分类序】选项卡查看控件的属性，也可以在属性窗口中编辑对象的属性。

- 代码窗口：其功能是编辑和存放 VBA 代码，相当于文字编辑器。
- 用户窗体：在工程中创建窗口或对话框。可以画出并查看窗体上的控件。
- 立即窗口：可以对测试的代码马上给出结果，供程序设计者参考。
- 本地窗口：可自动显示出所有在当前过程中的变量声明及变量值。

【提示】在实际开发的过程中，VBA 窗口界面中所有的功能窗口不一定都同时显示出来，用户可以根据开发习惯进行修改界面，例如可以隐藏不常用的子窗口，还可以更改子窗口的大小和位置等操作。



2.2 VBE 的子窗口

了解了 VBE 的窗口后，下面继续详细学习 VBE 的子窗口。

2.2.1 工程资源管理窗口

在 VBA 的【工程】窗口中，可以把每一个打开的 Excel 工作簿看作一个工程，且工程的默认名称为 VBAProject。一个新建的工作簿只包含 Excel 对象，如图 2-7 所示。

如果工程中包含 VBA 模块或用户窗体等，那么工程列表中同样会显示出相应的节点，如图 2-8 所示。



图 2-7 默认【工程】窗口

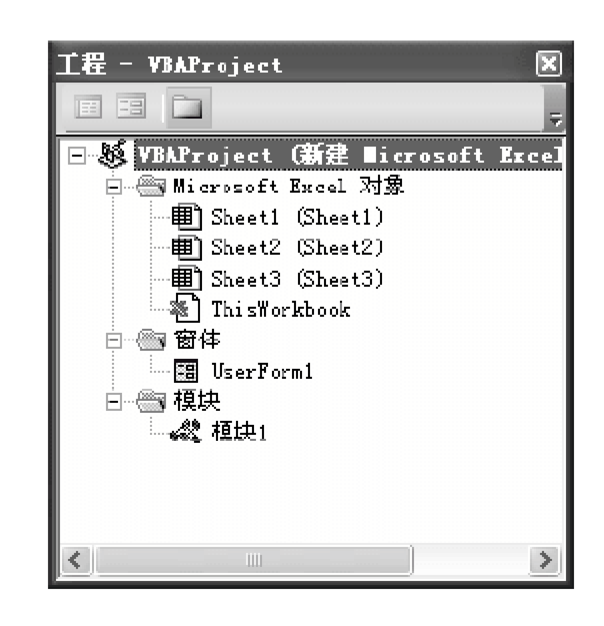


图 2-8 包含模块和窗体的【工程】窗口

如果在打开的 VBA 窗口中没有显示【工程】窗口，可以选择【视图】➢【工程资源管理器】菜单命令或按 Ctrl+R 组合键将其打开。

【提示】当打开 VBA 时，不能认定当前显示的代码窗口就是【工程】窗口中高亮显示的对象所对应的程序窗口，为确保在正确的代码窗口中编辑 VBA 代码，通常需要在【工程】窗口中双击需要编辑 VBA 代码的对象。

如果用户想要在【工程】窗口中进行插入模块、插入窗体、导入文件或导出文件等操作，

可以在选中相应的对象工程名称后，通过单击鼠标右键，在弹出的快捷菜单中选择相应的菜单命令来完成。

另外，在【工程】窗口中可以删除 VBA 模块，但不能移除与工作簿或工作表相关联的代码模块。

2.2.2　代码窗口

工程中的每一个对象都有一个相关联的代码窗口，该窗口的主要作用是编辑和存放程序。每个对象的代码窗口都是由【对象】下拉列表框、【过程/事件】下拉列表框、【过程视图】图标和过程编辑区等部分组成的，如图 2-9 所示。

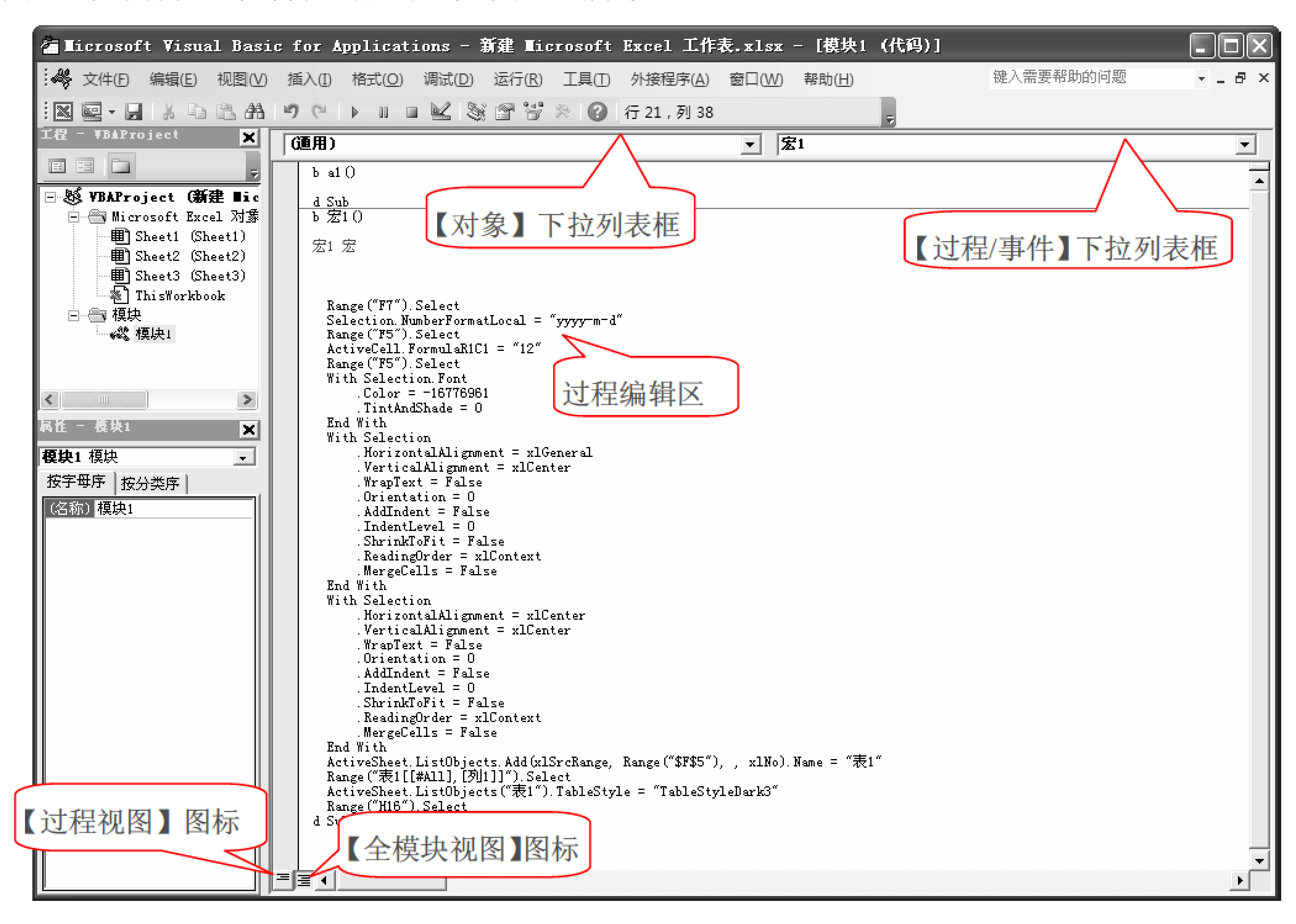


图 2-9　代码窗口

其中【对象】下拉列表中会显示所选对象的名称。【过程/事件】下拉列表中会列出指定对象控件显示所选的程序，并且在程序代码窗口中同一时间只能显示一个程序，但是【全模块视图】模式在同一时间可以显示多个程序。

如果要查看某个对象的代码窗口或要在指定的对象中编写程序，只需在【工程】窗口中双击该对象即可打开其代码窗口。

2.2.3　用户窗体

用户窗体是显示在应用程序中的对话框，它是 VBA 中一个非常重要的组成部分。

1. 插入用户窗体

窗体的添加可以使用户有更多的机会与程序对话。插入用户窗体，除了在【工程资源管理器】窗口中单击鼠标右键的方法外，用户还可以通过以下方法来实现。

（1）在 VBA 编辑窗口中选择【插入】➢【用户窗体】菜单命令，即可插入用户窗体。

（2）在【标准】工具栏上单击【插入用户窗体】按钮右侧的下三角按钮，在其下拉列表中选择【用户窗体】选项即可。

以上两种方法都可以插入用户窗体，新插入的用户窗体的默认名称为 UserForm1、UserForm2 等。另外，在插入用户窗体的同时系统还会自动打开控件工具箱，如图 2-10 所示。

2. 更改窗体名称

为了更容易识别用户窗体，可以在其对应的【属性】窗口中切换到【按字母序】选项卡，在【名称】文本框中输入需要设置的名称，即可更改【工程】窗口中该用户的显示名称。而在【Caption】文本框中输入新的名称后即可更改用户窗体标题栏的名称，如图 2-11 所示。

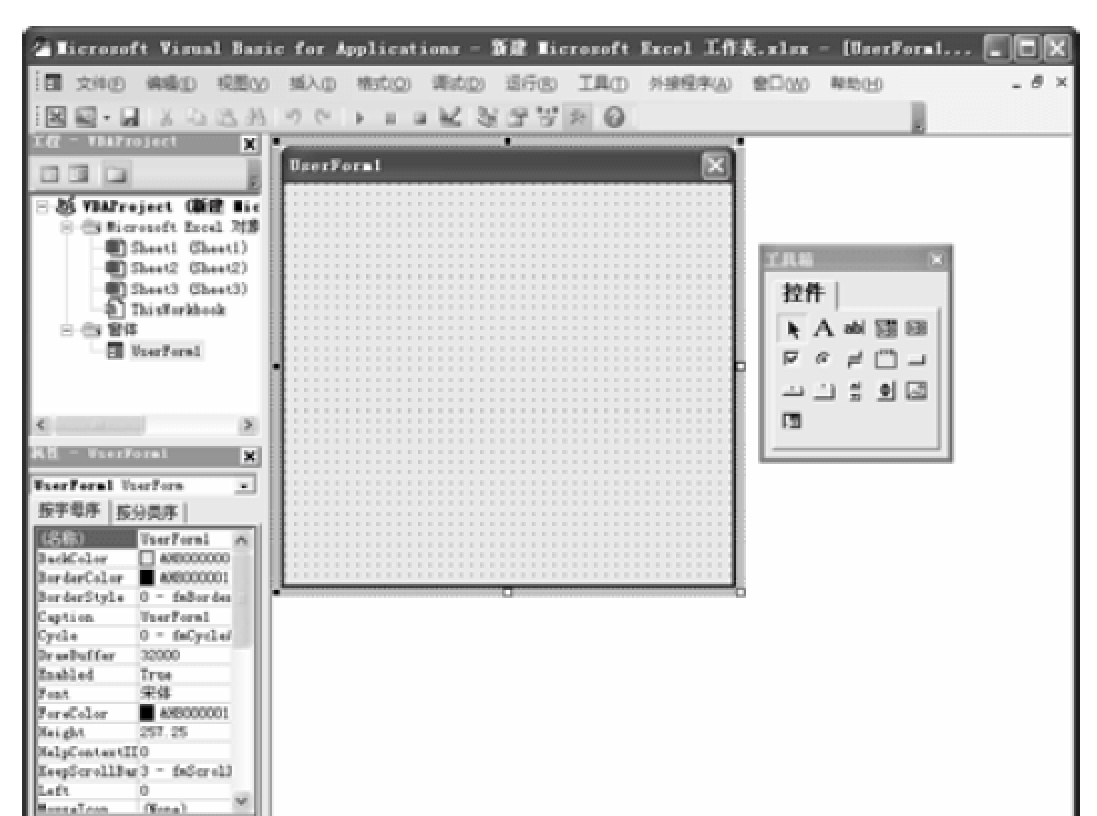


图 2-10　新建的用户窗体

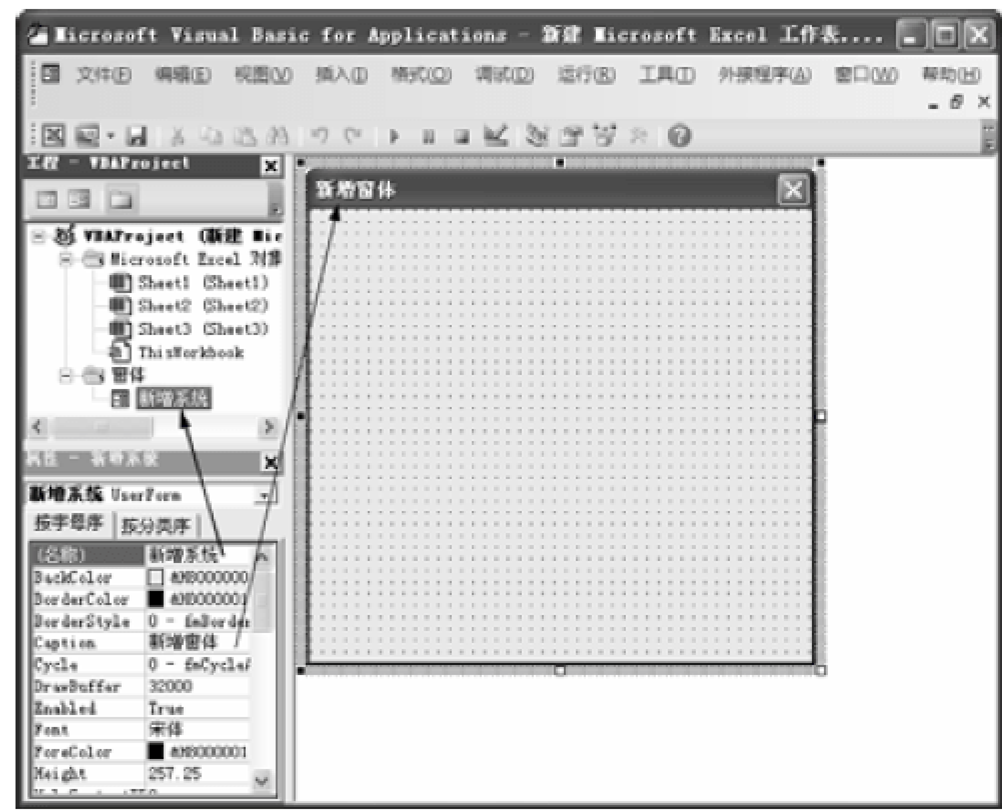


图 2-11　更改用户窗体的名称

3. 利用控件工具箱

可以利用控件【工具箱】中的【命令按钮】按钮、【文字框】按钮和【标签】按钮等，在用户窗体上添加相应的控件，如图 2-12 所示。

4. 移除用户窗体

在【工程】窗口中选择要移除的用户窗体，如这里要移除名称为“搜索”的窗体，单击鼠标右键，在弹出的快捷菜单中选择【移除 搜索】菜单命令，系统即可弹出一个提示对话框，单击【否】按钮即可将其移除，如图 2-13 所示。



图 2-12　【工具箱】对话框



图 2-13　【移除 搜索】的提示对话框

2.2.4　属性窗口

通过【属性】窗口不仅能够更改用户窗体的名称，还可以设置窗体的背景色、添加背景图片或设置图片的显示效果等。

下面通过更改属性窗口的参数为例进行讲解，具体操作步骤如下。

01 选择需要修改属性的窗体，在用户窗体对应的【属性】窗口中切换到【按字母序】选项卡，然后单击 BackColor（背景颜色）文本框右侧的下三角按钮，在弹出的面板中选择【调色板】选项卡，如图 2-14 所示。

02 选择合适的颜色后，即可发现窗体的背景发生了变化，如图 2-15 所示。



图 2-14　【调色板】选项卡

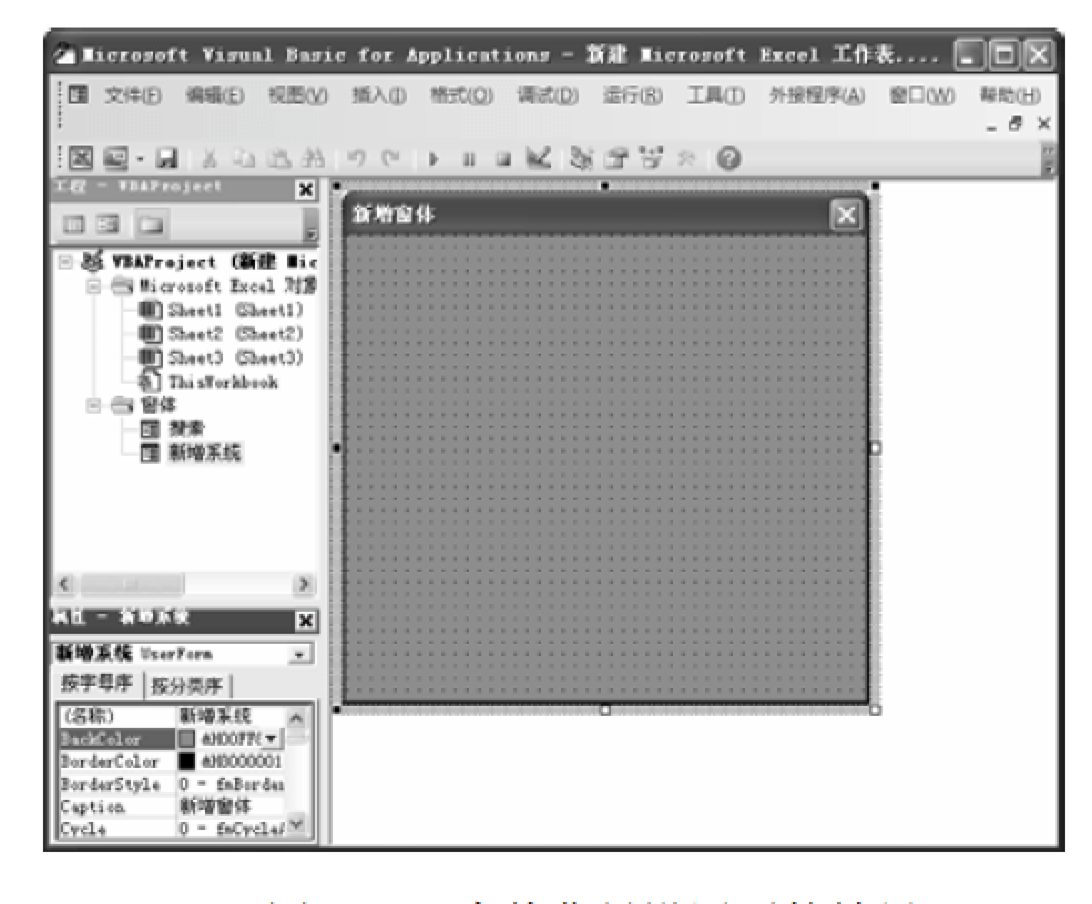


图 2-15　窗体背景设置后的效果

03 如果添加的为背景图片，则可在【属性】窗口中设置图片的显示效果。单击【Picture】文本框右侧的【浏览】按钮 ...，如图 2-16 所示。

04 弹出【加载图片】对话框，选择需要显示的图片文件，单击【打开】按钮，如图 2-17 所示。

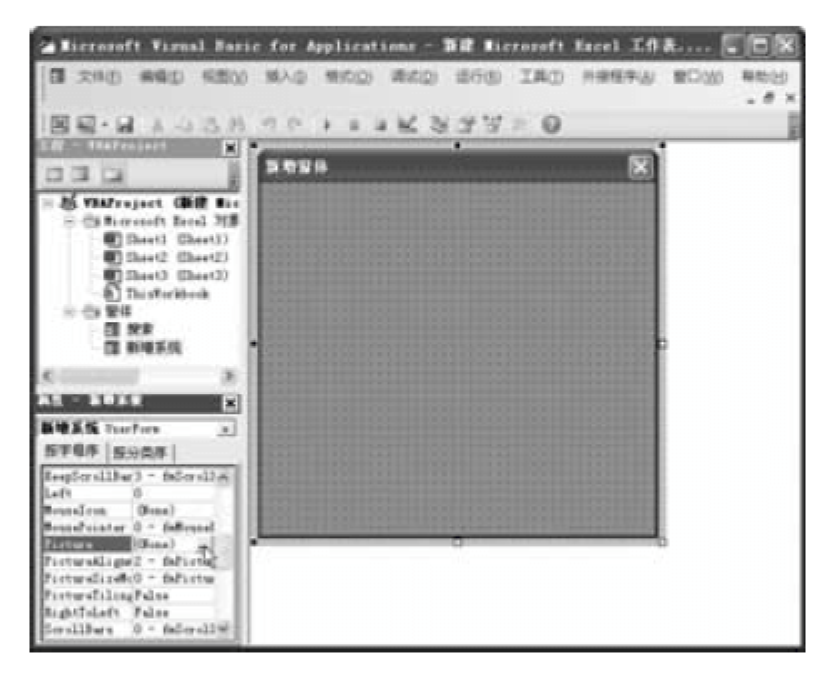


图 2-16　单击【浏览】按钮

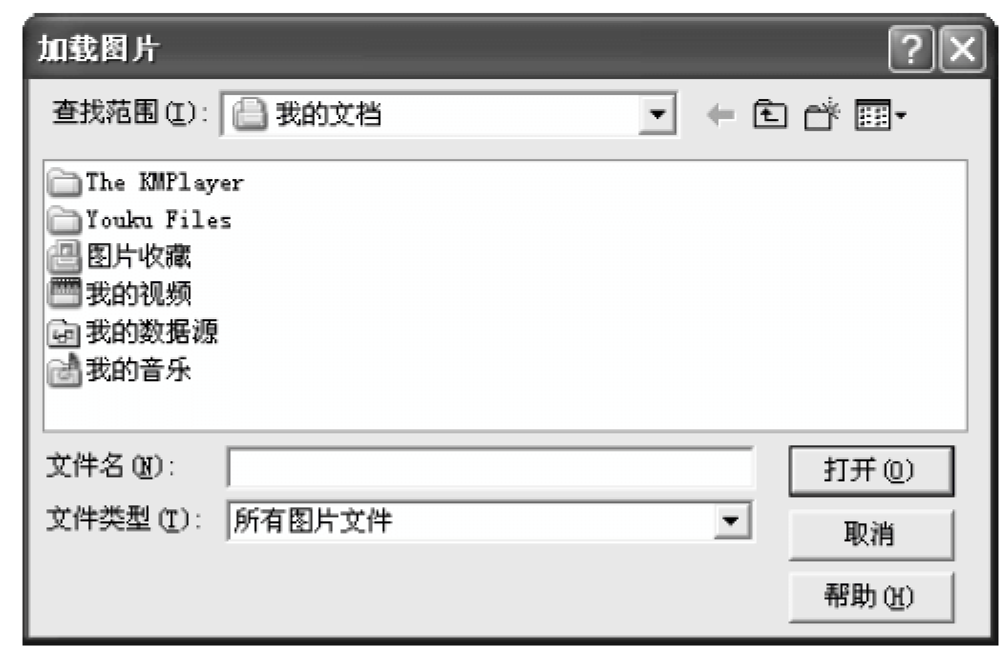


图 2-17　【加载图片】对话框

05 即可将图片加载为窗体背景，然后在 PictureAlignment 文本框中设置图片的对齐方式，在【PictureSizeMode】文本框中设置背景图片的显示方式，在【PictureTiling】文本框中设置图片在窗体中是否平衡显示等效果，如图 2-18 所示。

在用户窗体上双击鼠标即可打开用户窗体的代码窗口。也可以在【工程】窗口中进行切换，即单击【查看对象】按钮，可以由代码窗口转换为用户窗体界面，单击【查看代码】按钮，可以由用户窗体界面转换为代码窗口，如图 2-19 所示。

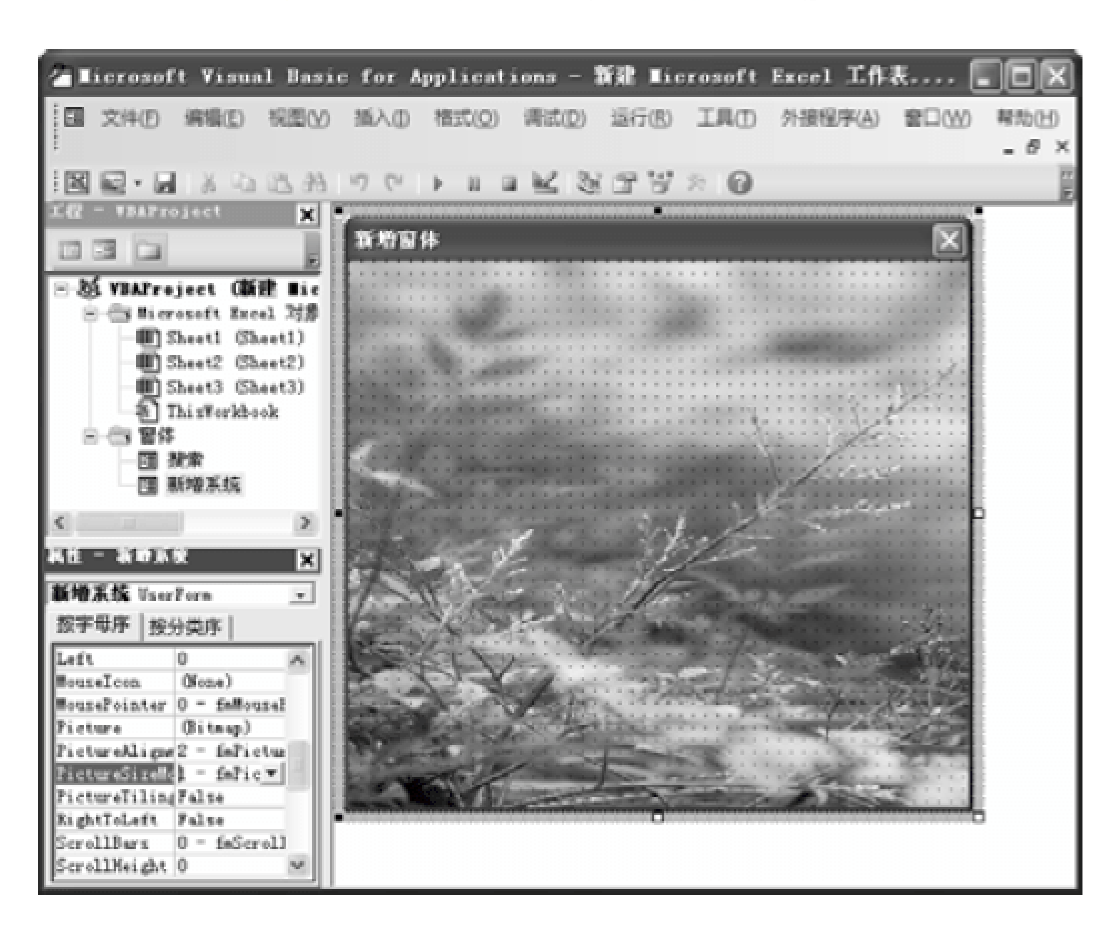


图 2-18　设置窗体背景

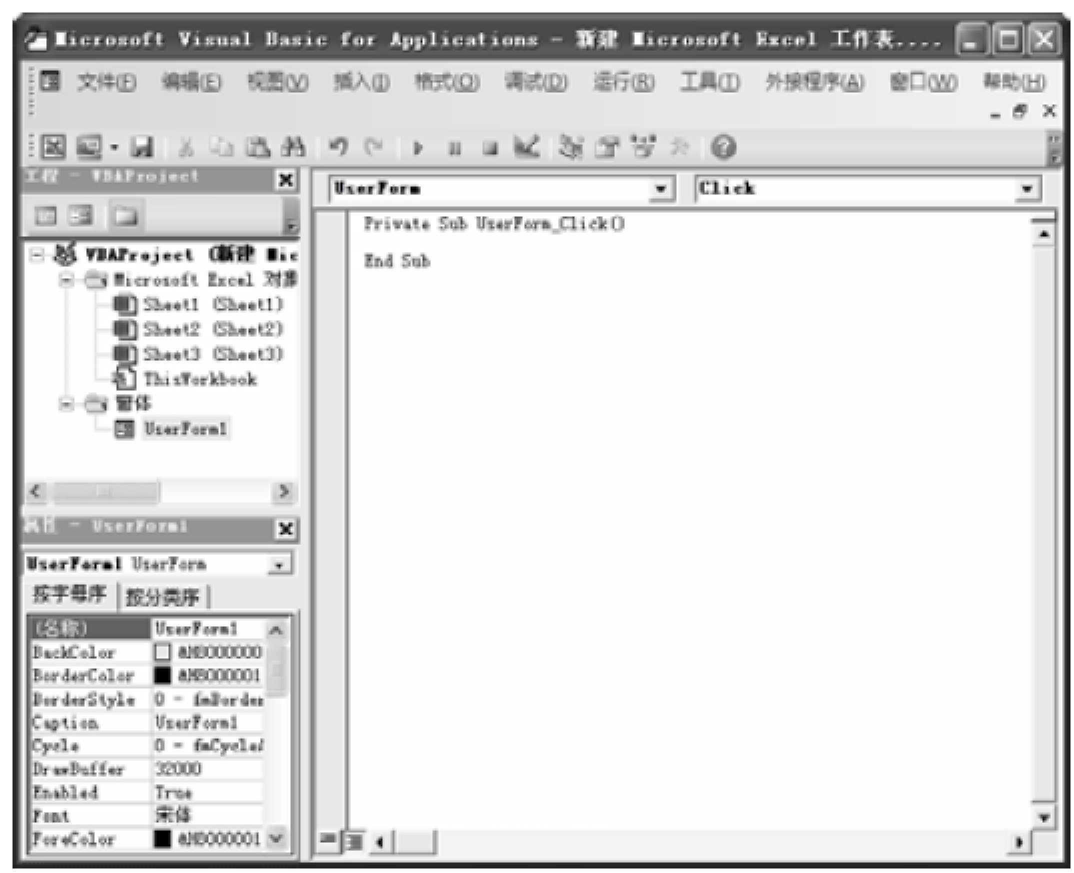


图 2-19　窗体的代码窗口



2.3 定制 VBE 环境

对于 Excel 开发人员来说，还应该掌握如何定制 VBE 环境，即根据用户的习惯和需要来设置 Visual Basic 编辑器环境选项。

下面以定制 VBE 环境为例进行讲解，具体操作步骤如下。

01 打开 VBA 编辑器窗口，选择【工具】➤【选项】菜单命令，如图 2-20 所示。

02 弹出【选项】对话框，该对话框共有 4 个选项卡：【编辑器】、【编辑器格式】、【通用】和【可连接的】。在【编辑器】选项卡中可以定制代码窗口，包括代码设置和窗口设置，如图 2-21 所示。



图 2-20　选择【选项】菜单命令

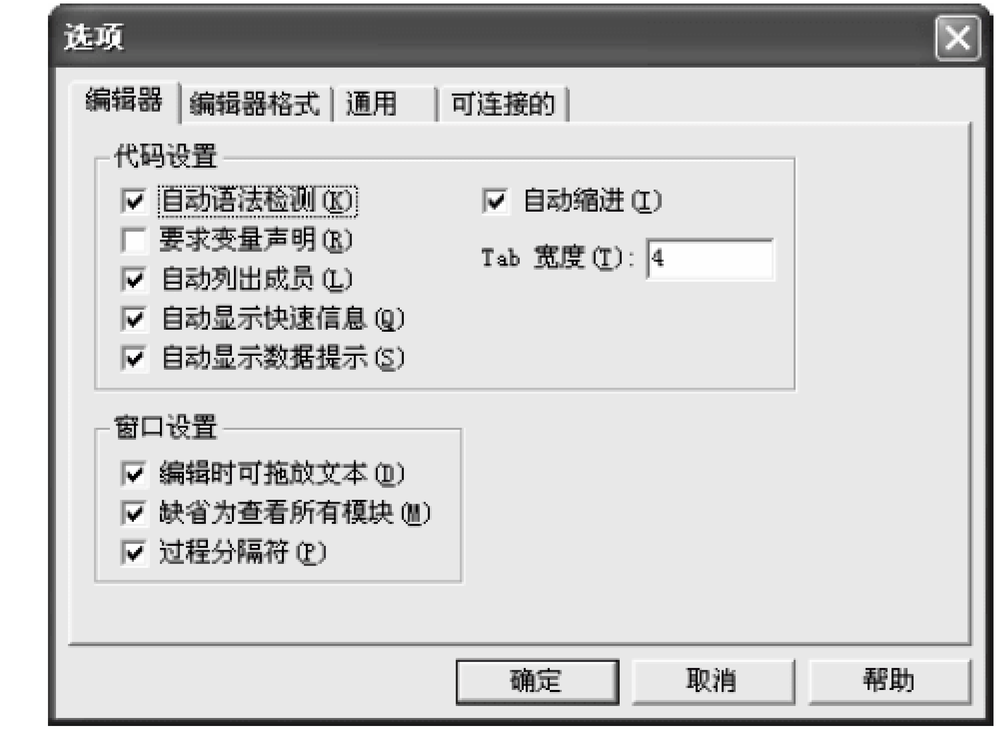


图 2-21　【编辑器】选项卡

【选项】对话框中各个参数的含义如下。

- 【自动语法检测】：选择该复选框后，在输入一行代码后，系统会自动检测语法。如果未选择此复选框，VBE 通过使用不同的颜色来显示语法错误的代码，此时不会弹出提示对话框。
- 【要求变量声明】：选择该复选框后，VBE 将会在新插入的模块开始处增加语句“Option Explicit”，一旦出现此语句，必须在定义模块中使用每个变量，否则将出现错误提示。
- 【自动列出成员】：选择该复选框，在输入代码时，VBE 将自动列出对象的成员列表。
- 【自动显示快速信息】：选择该复选框后，将显示所输入函数和参数的信息。
- 【自动显示数据提示】：选择该复选框后，将显示出指针所在位置的变量值。
- 【自动缩进】：选择该复选框，VBE 将按照 Tab 宽度自动缩进显示每行代码。
- 【Tab 宽度】：在文本框中输入缩进量，默认值是 4 个空格，设置范围为 1~32 个空格。
- 【编辑时可拖放文本】：选择该复选框后，将允许通过拖动操作来复制和移动代码窗口中的文本。
- 【默认为查看所有模块】：选择该复选框后，在代码窗口中可以查看模块的所有过程。
- 【过程分隔符】：选择该复选框后，即可在每个过程尾端显示分隔符条。

03 选择【编辑器格式】选项卡，在【代码颜色】列表中选择【标准文本】选项，然后设置前景色、背景色和标识色，在【字体】列表中设置字体和字号，在【示例】中即可看到预览效果，如图 2-22 所示。

04 选择【通用】选项卡，可以设置 VBE 的工程设置、错误处理和编译设置。一般情况下，采用默认的设置即可，如图 2-23 所示。



图 2-22 【编辑器格式】选项卡

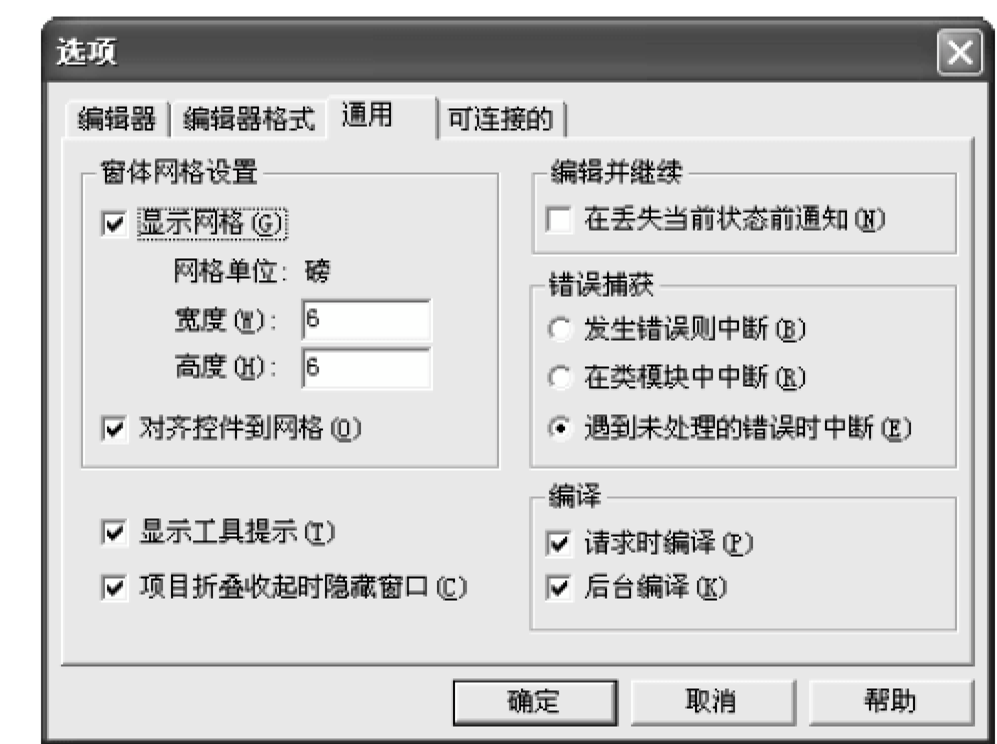


图 2-23 【通用】选项卡

05 选择【可连接的】选项卡，用户可以选出要连接的窗口，如图 2-24 所示。连接发生在当窗口附加到其他可连接的窗口或应用程序窗口的边缘时。当移动一可连接的窗口时，该窗口很快地移向此位置上。

06 设置完成后，单击【确定】按钮，即可看到定制的 VBE 环境，如图 2-25 所示。



图 2-24 【可连接的】选项卡

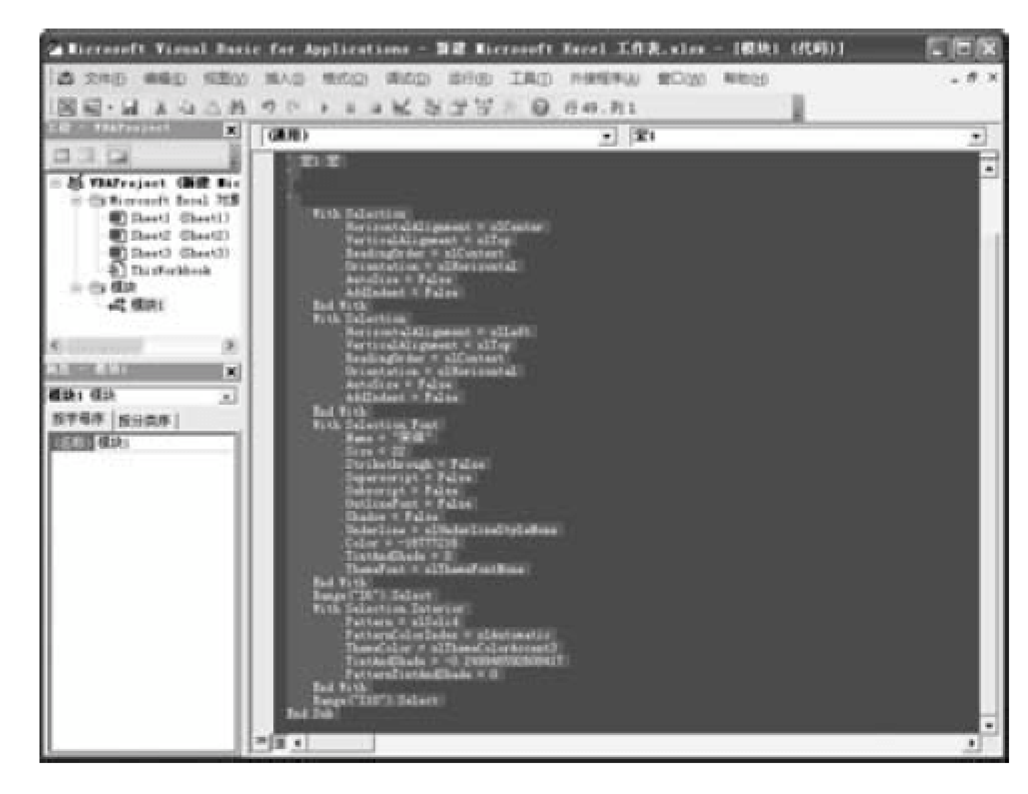


图 2-25 定制的 VBE 编辑器窗口



2.4 使用帮助

使用帮助是学习 Excel VBA 最快捷的方法。Excel 提供的帮助程序非常详细，并提供了许多例子供初学者学习。和普通的帮助程序相比，Excel VBA 的帮助文档更加人性化，可以帮助用户快速地找到需要查找的内容。

使用帮助的具体操作步骤如下。

01 打开 VBA 编辑器窗口，将光标放在需要查看帮助的代码的某个对象、方法或属性，例如放在 Sub 代码上，按 F1 快捷键，如图 2-26 所示。

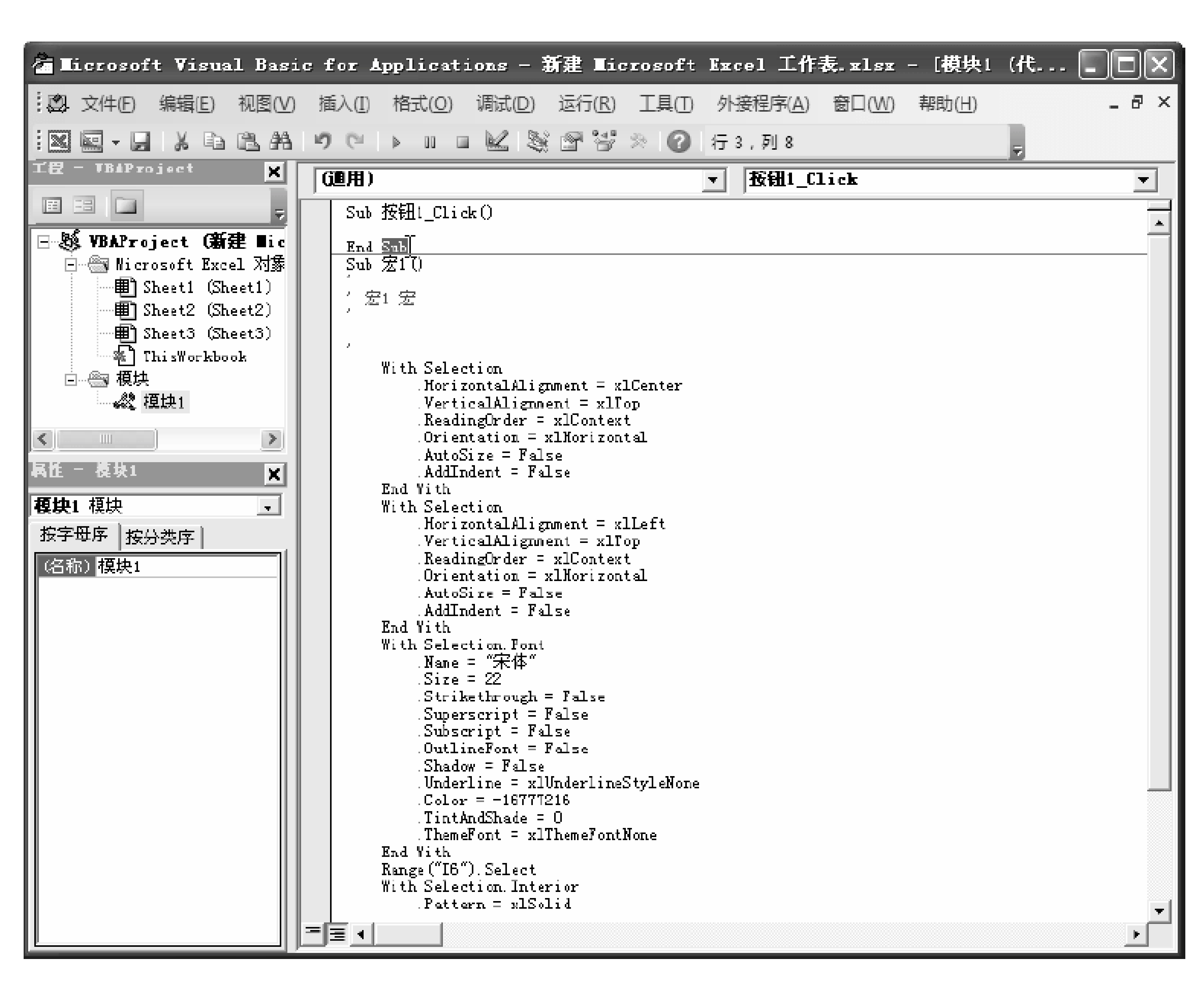


图 2-26　鼠标放在 Sub 代码上

02 弹出 Excel 帮助窗口，并显示 Sub 语句的相关帮助文档，用户可以查看 Sub 的使用方法，如图 2-27 所示。

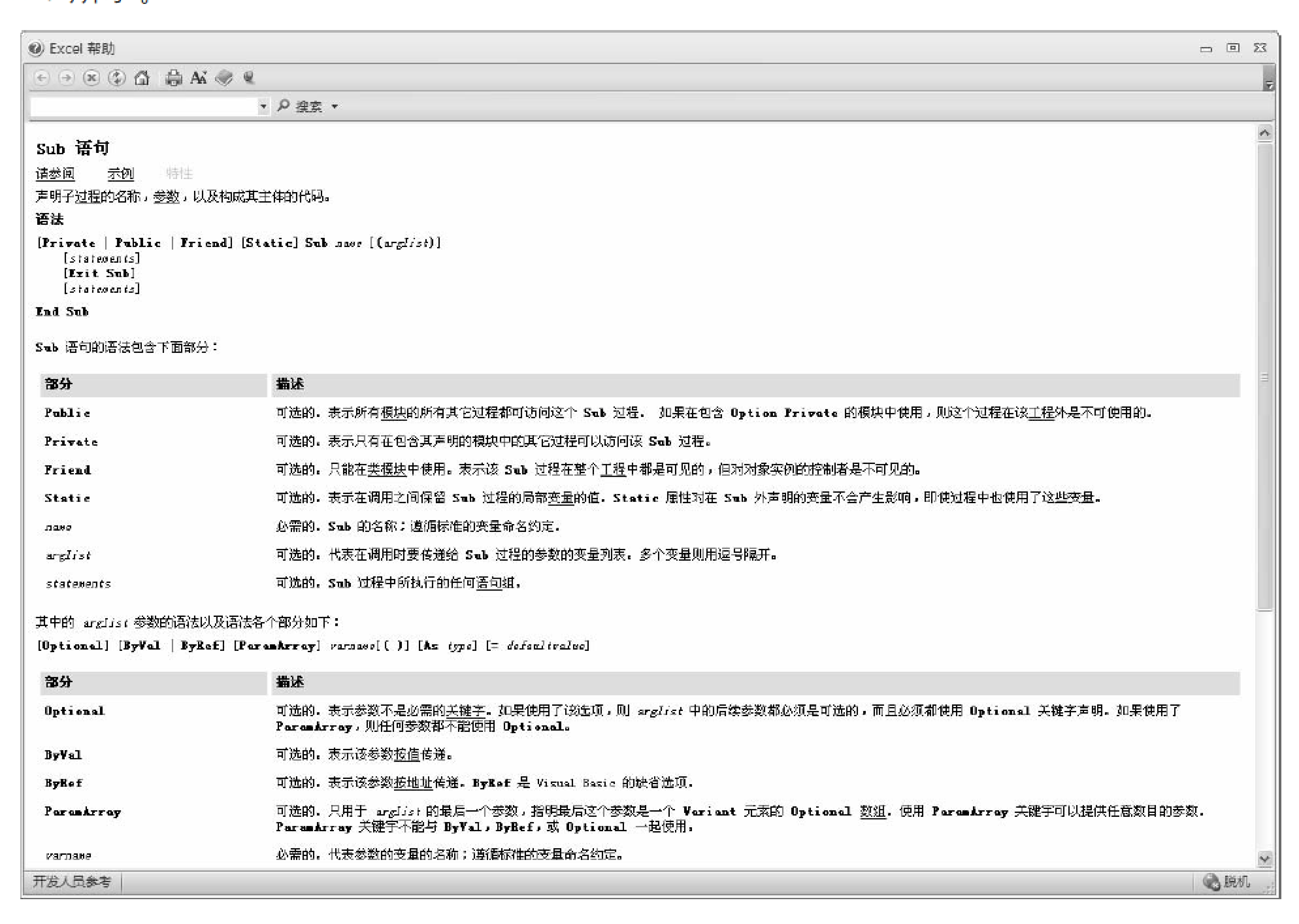


图 2-27　Excel 帮助窗口

03 如果用户想浏览帮助文档，并系统学习 VBA 的相关知识，可以直接按 F1 快捷键，打开

帮助的起始窗口，单击窗口中的相关链接，即可查看帮助文档的内容，如图 2-28 所示。



图 2-28　系统默认帮助窗口

04 在帮助文档窗口中，用户也可以在搜索框中直接输入需要搜索的关键词，例如输入 sub，单击【搜索】按钮，即可显示相关的搜索结果列表，如图 2-29 所示。



图 2-29　输入 sub

05 选择需要查看内容的超链接，例如"Sub 语句"，即可打开具体的帮助内容。如果想查看 Sub 语句的相关实例，可以单击【示例】超链接，如图 2-30 所示。



图 2-30　“Sub 语句”的帮助内容

06 即可打开示例窗口，用户可以复制例子的代码到代码窗口中运行，可以通过学习实例就可以掌握相关代码的编写方法，如图 2-31 所示。



图 2-31　VBE 编辑器窗口



2.5 高手私房菜

本章主要介绍了 Excel 2010 VBA 的开发环境，下面继续来解答一些常见的疑惑。

技巧 1：快速给工程加密

如果不想让别人看到工程的具体细节，可以锁定工程并添加密码。具体操作步骤如下。



01 进入 VBE 操作界面，选择【工具】➢【VBAProject 属性】菜单命令，如图 2-32 所示。

02 弹出【VBAProject-工程属性】对话框，选中【查看时锁定工程】复选框，然后在【密码】和【确认密码】文本框中输入工程属性密码，单击【确定】按钮即可。

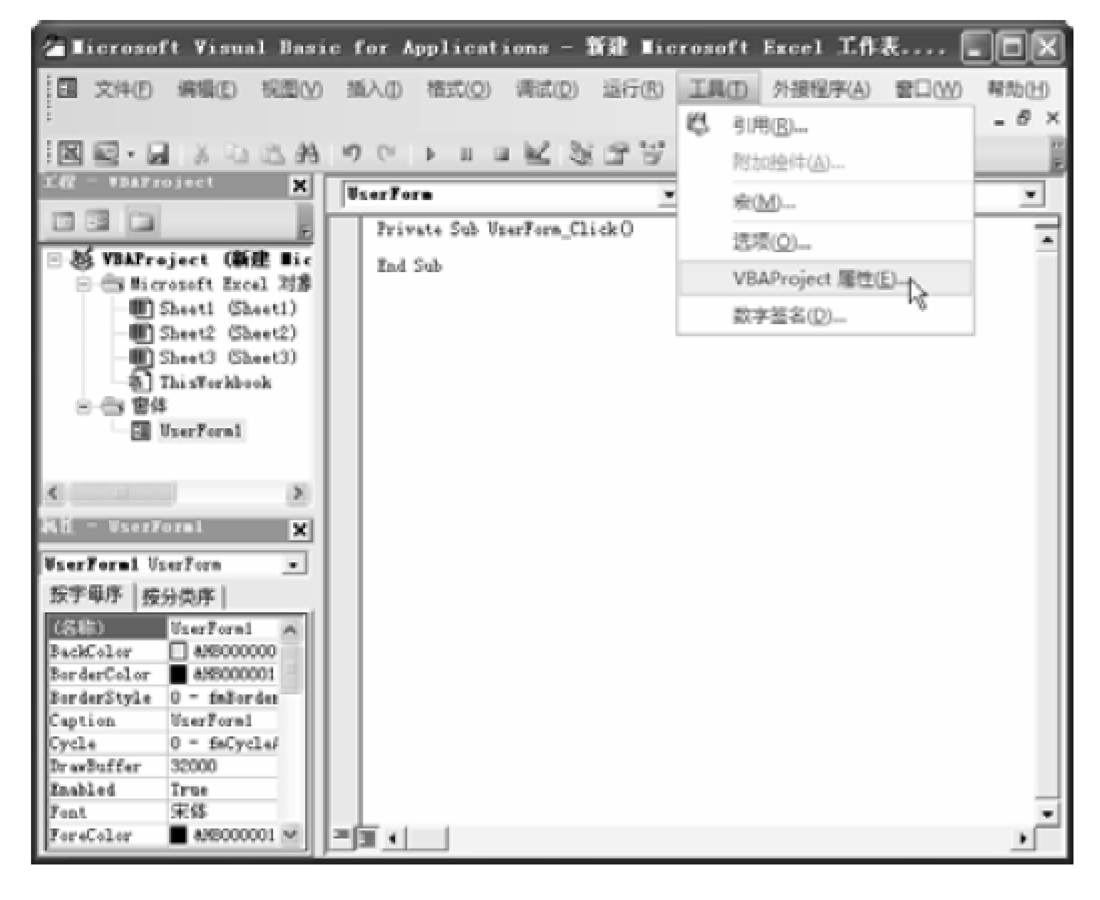


图 2-32 选择【VBAProject 属性】菜单命令

VBAProject - 工程属性

通用 保护

锁定工程

☑ 查看时锁定工程 (V)

查看工程属性的密码

密码 (P) ******

确认密码 (C) ******

确定 取消 帮助

图 2-33 【VBAProject-工程属性】对话框

技巧 2：找回丢失的工具栏

如果用户不小心关闭或者隐藏了控件【工具箱】窗口，可以选择【视图】➢【工具箱】菜单命令，或单击【标准】工具栏上的【工具箱】按钮将其重新调出。

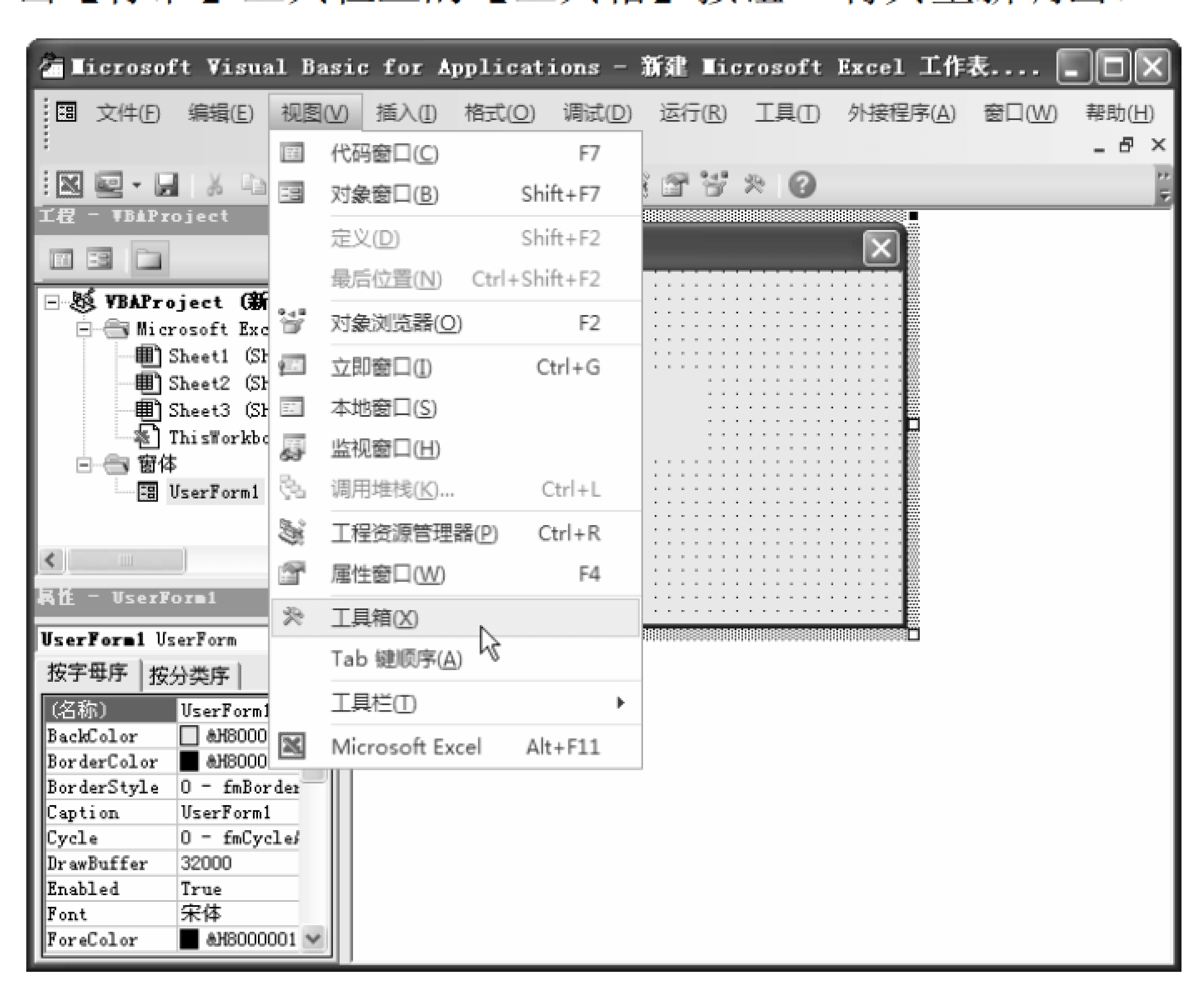


图 2-34 选择【工具箱】菜单命令

第 3 章　宏概述

学习完了 VBA 应用程序开发的平台和环境后，开始学习如何使用 Excel 创建和使用宏，这也是 Excel 中较为强大的功能。本章主要讲述宏的基本概念、创建宏的各种方法、如何管理宏、运行宏和设置宏的安全性，最后通过 1 个综合案例，讲述宏的具体录制方法。



3.1 宏的基本概念

在学习 Excel VBA 时，不能不提到 Excel 中的宏，这里将对宏的概念和用途等进行介绍，以便于用户能够更好地理解宏的含义。

3.1.1 什么是宏

宏是通过一次单击就可以应用的命令集。它们几乎可以自动完成用户在程序中执行的任何操作，甚至还可以执行用户认为不可能的任务。在 Excel 的【视图】选项卡中可以单击【宏】按钮，弹出的下拉菜单中包含常见的宏操作，如图 3-1 所示。

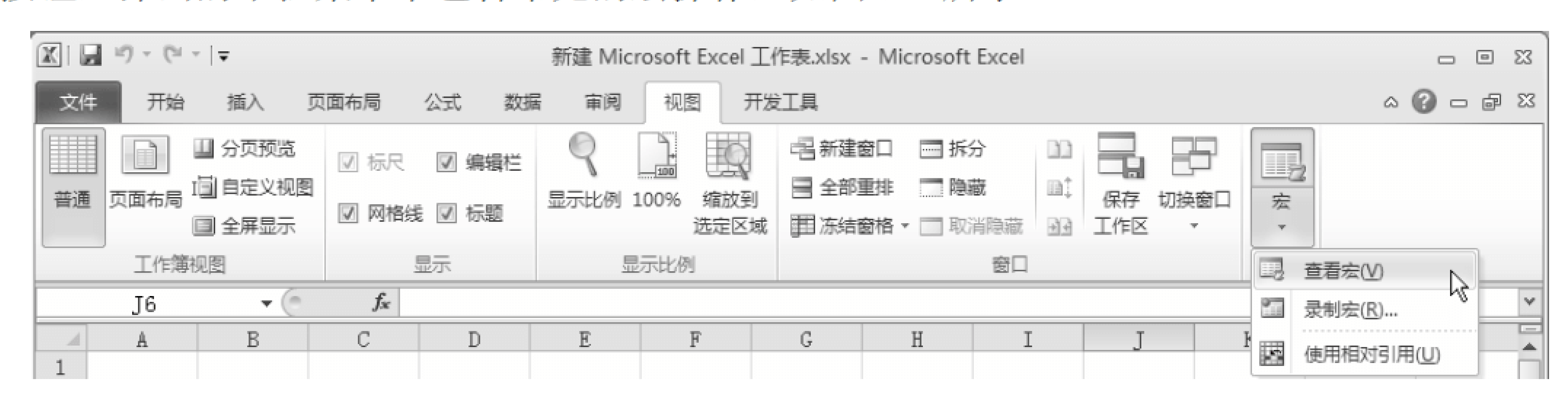


图 3-1　【视图】选项卡中的【宏】按钮

由于工作需要，每天都在使用 Excel 进行表格的编制、数据的统计等，每一种操作可以称为一个过程。而在这个过程中，经常需要进行很多重复性操作，如何能让这些操作自动重复执行呢？Excel 中的宏恰好能解决这类问题。

宏不仅可以节省时间，并可以扩展日常使用的程序的功能。使用宏可以自动执行重复的文档制作任务，简化繁冗的操作，还可以创建解决方案。VBA 高手们可以使用宏创建包括模板、对话框在内的自定义外接程序，甚至可以存储信息以便重复使用。

从更专业的宏来说，宏是保存在 Visual Basic 模块中的一组代码。正是这些代码驱动着操作的自动执行。当单击按钮时，这些代码组成的宏就会执行代码记录的操作，如图 3-2 所示。

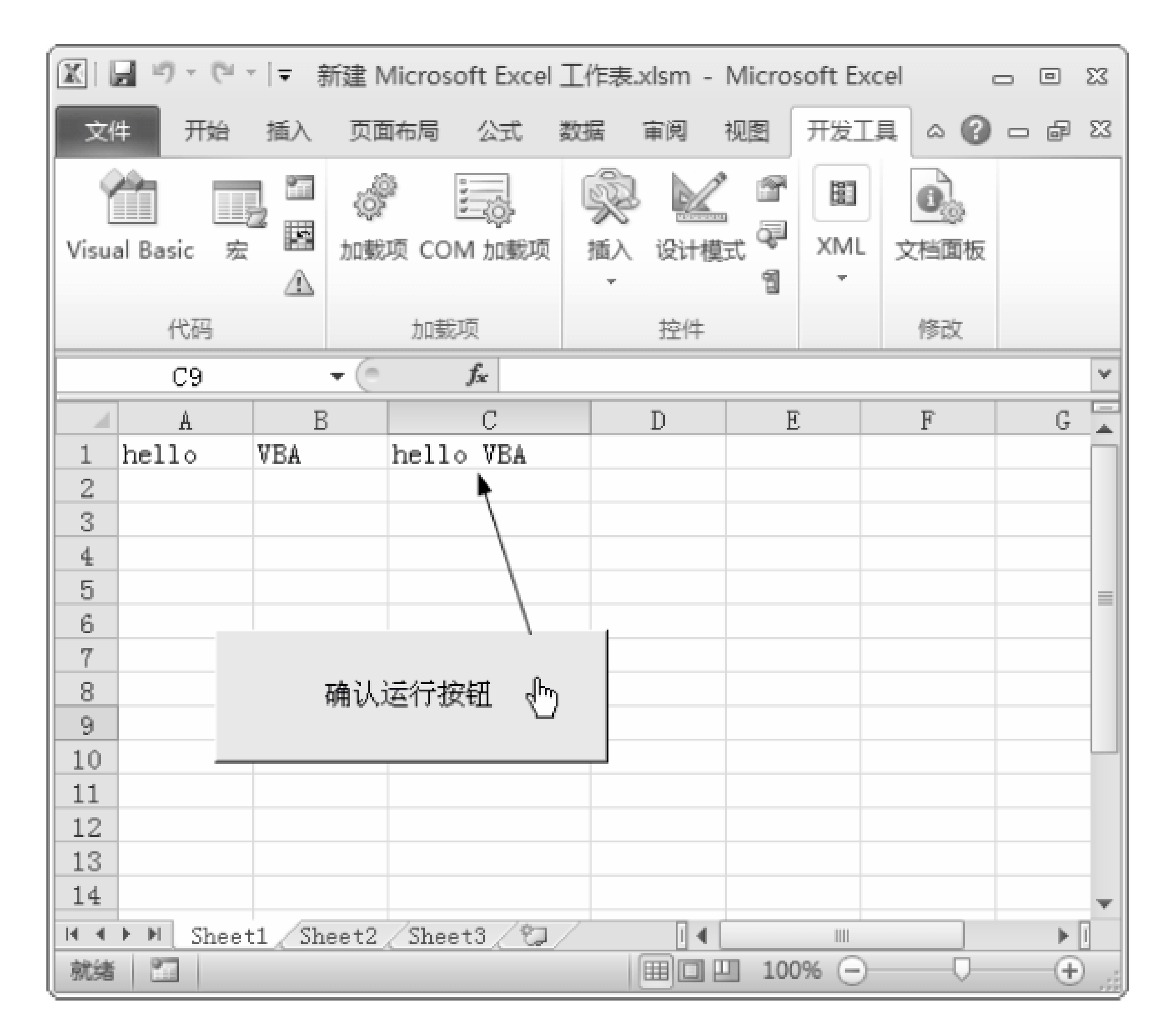


图 3-2 单击按钮执行宏操作

单击【开发工具】选项卡中【代码】选项组中的 Visual Basic 按钮，即可打开 VBA 的代码窗口，用户可以看到宏的具体代码，如图 3-3 所示。

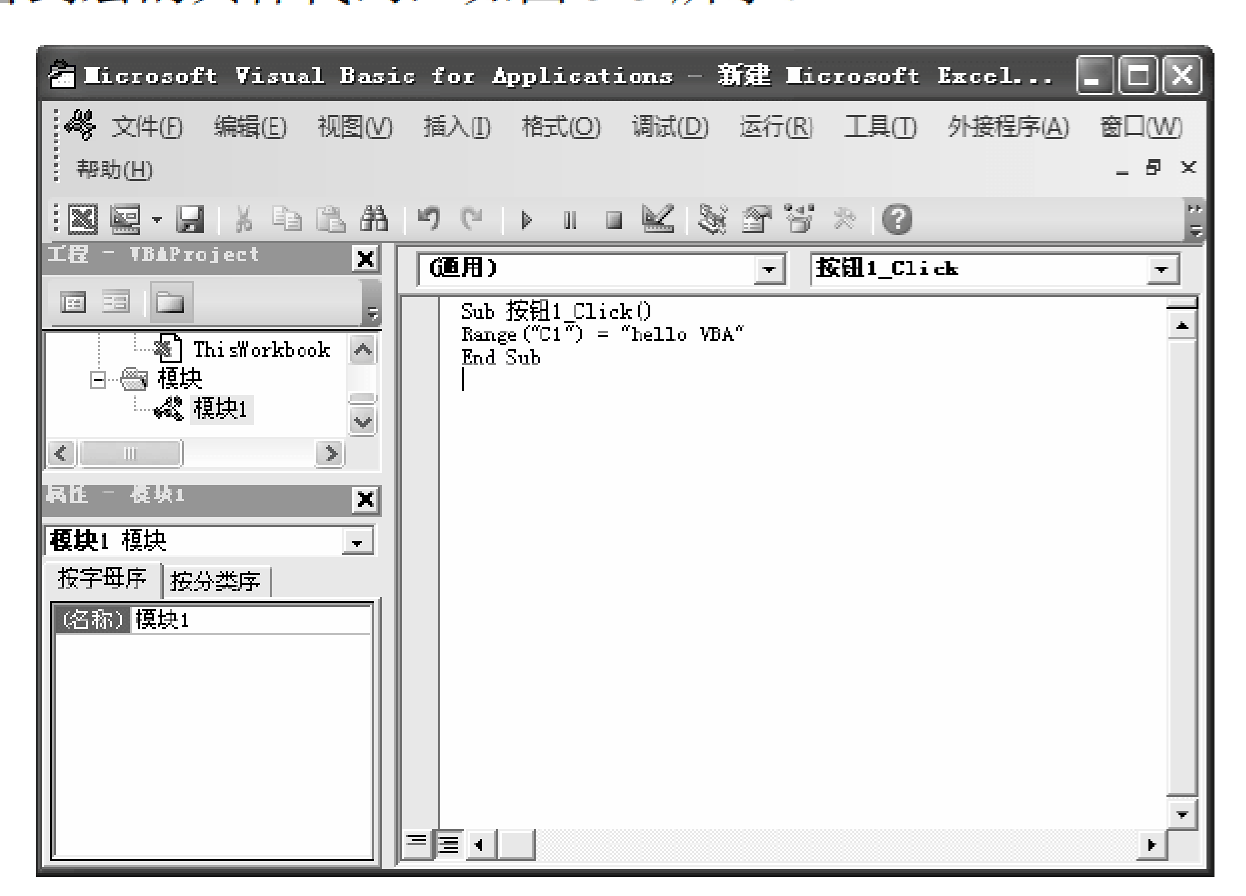


图 3-3 宏的代码

【提示】在上述宏代码中“按钮 1_Click”是宏的名称。语句 Range(“C1”)= “hello VBA”是用代码方式表示的操作过程，即在活动单元格 C1 中录入英文 hello VBA。

3.1.2 宏的作用

在实际的工作中，使用宏可以实现复杂的工作简单化，重复的工作便捷化，从而大幅度

提高工作效率。其常见应用如下。

（1）将重复的工作定义成模块，利用按钮功能来方便操作。

工作实际上是由若干件重复的工作组来实现的。可以将大量的、重复的工作组定义成一些模块，并且用设定的按钮来进行控制和操作，这样可以简化一些大量重复工作的操作过程，从而提高工作效率。

（2）可以根据业务需要，进行操作界面的定制，使 Excel 环境成为一个业务系统。

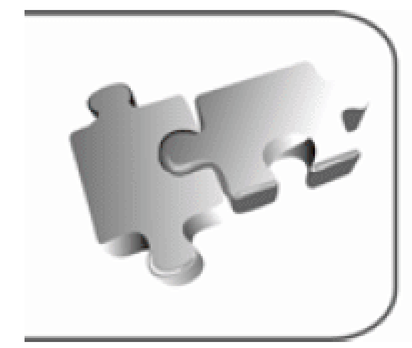
可以利用 VBA 就我们的业务系统和一些特殊功能界面进行一些简单的定制，从而使 Excel 成为一种虚拟的业务系统。比如说，我们经常要大量地录入数据，是利用 Excel 进行客户资料的管理，在录入这些数据的时候，要进行不断的、重复的、切换性的操作，这样会浪费大量的鼠标和键盘操作，但是我们可以利用 Excel VBA 来定制开发出这样一个管理环境，从而在一个简单的界面下完成所有的客户资料管理。

（3）创建报表系统，定制开发系统报表功能，简化在实际报表设计中的操作过程。

销售和财务部门需要大量的报表系统，在日常 Excel 操作时所使用的报表系统主要有图表和数据透视表，其他的都是通过单元格的合并和其他一些操作来实现的，现在呢，可以利用 VBA 根据工作的需要设计和定制一套全面的报表系统，并且在这个系统当中，可以根据需要加以设计，从而达到一个比较科学的、方便的生成系统。

（4）创建工作需要的特殊计算公式。

Excel 一个强大的功能就是函数公式，但是有一些功能没有设计到公式中去。例如，在 Excel 中有一个连加公式，但在实际工作中需要连减公式就没有了。又例如，在 Excel 中进行财务处理，有很多的财务公式，但是缺乏的是从小写字母转换成大写字母的公式，在国内企业中，财务部甚至许多部门都需要使用到这个公式。那怎么办呢？这时，就可以通过 VBA 进行开发和定制一个从小写模式转换或大写模式的公式，从而提高我们的输入速度和效率。



3.2 创建宏

了解了宏的基本概念后，下面开始学习如何创建宏。

3.2.1　添加【开发工具】选项卡

创建宏的过程中，需要用到 Excel 2010 的【开发工具】选项卡。默认情况下，【开发工具】选项卡并不显示。下面讲述如何添加【开发工具】选项卡，具体操作步骤如下。

01 启动 Excel 2010，选择【文件】➤【选项】菜单命令，即可打开如图 3-4 所示的界面。

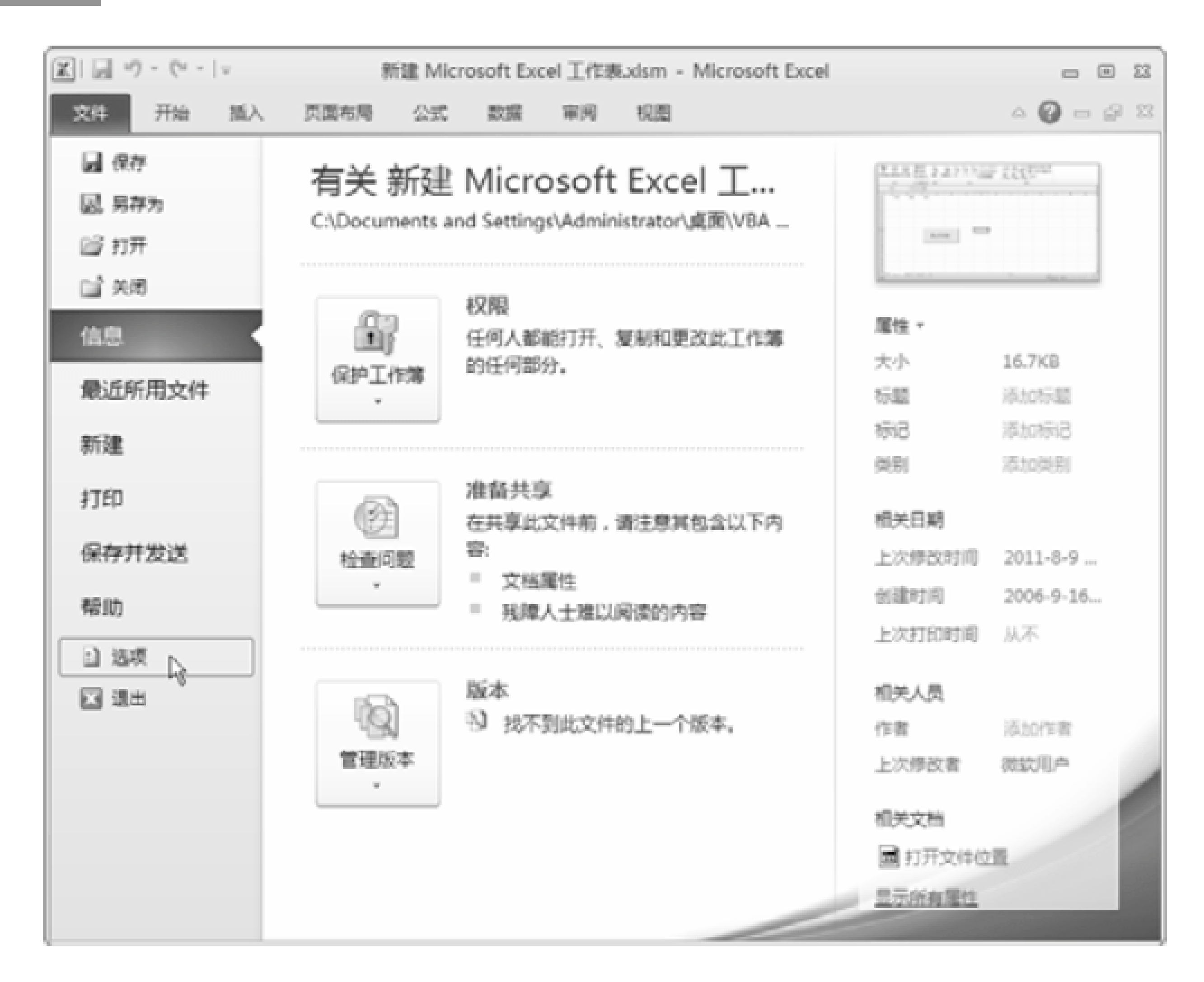


图 3-4 【文件】菜单

02 弹出【Excel 选项】对话框，在左侧列表中选择【自定义功能区】选项，在右侧的【自定义功能区】窗格中选中【开发工具】复选框，单击【确定】按钮，如图 3-5 所示。



图 3-5 【Excel 选项】对话框

03 即可成功显示【开发工具】选项卡，如图 3-14 所示。

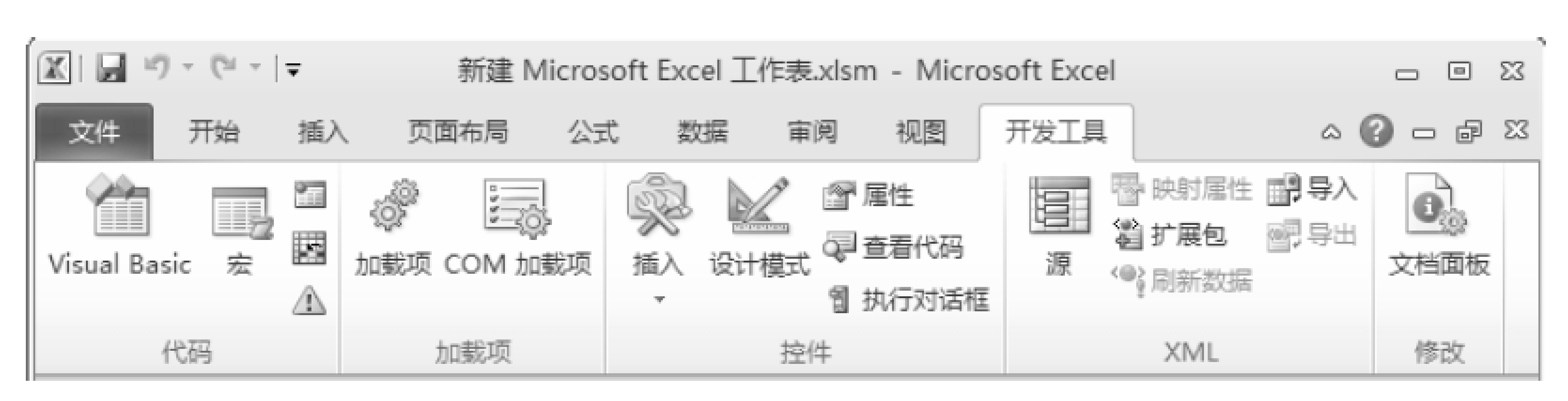


图 3-6 【开发工具】选项卡

3.2.2 创建宏的方法

在 Excel 中创建宏的方法有以下两种。

1. 利用宏录入器录制宏

下面以录制一个修改单元格底纹的实例进行讲解，具体操作步骤如下。

01 新建空白工作簿，选择 A1 单元格，选择【开发工具】选项卡，在【代码】选项组中单击【录制宏】按钮，如图 3-7 所示。

02 弹出【录制新宏】对话框，在【宏名】文本框中输入“修改底纹”，单击【确定】按钮，如图 3-8 所示。




图 3-7　单击【录制宏】按钮



图 3-8　【录制新宏】对话框

【提示】在【保存在】下拉列表框中共有 3 个选项：【当前工作簿】选项，表示只有当该工作簿打开时，录制的宏才可以使用；【新工作簿】选项，表示录制的宏只能在新工作簿中使用；【个人宏工作簿】选项，表示录制的宏可以在多个工作簿中使用。

03 右击 A1 单元格，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，如图 3-9 所示。

04 弹出【设置单元格格式】对话框。选择【填充】选项卡，然后设置背景颜色为红色、图案颜色为绿色，单击【确定】按钮，如图 3-10 所示。



图 3-9　【设置单元格格式】菜单命令



图 3-10　【设置单元格格式】对话框

05 即可看到 A1 单元格的底纹颜色发生了变化，单击【代码】选项组中【停止录制】按钮，即可完成宏的录制，如图 3-11 所示。

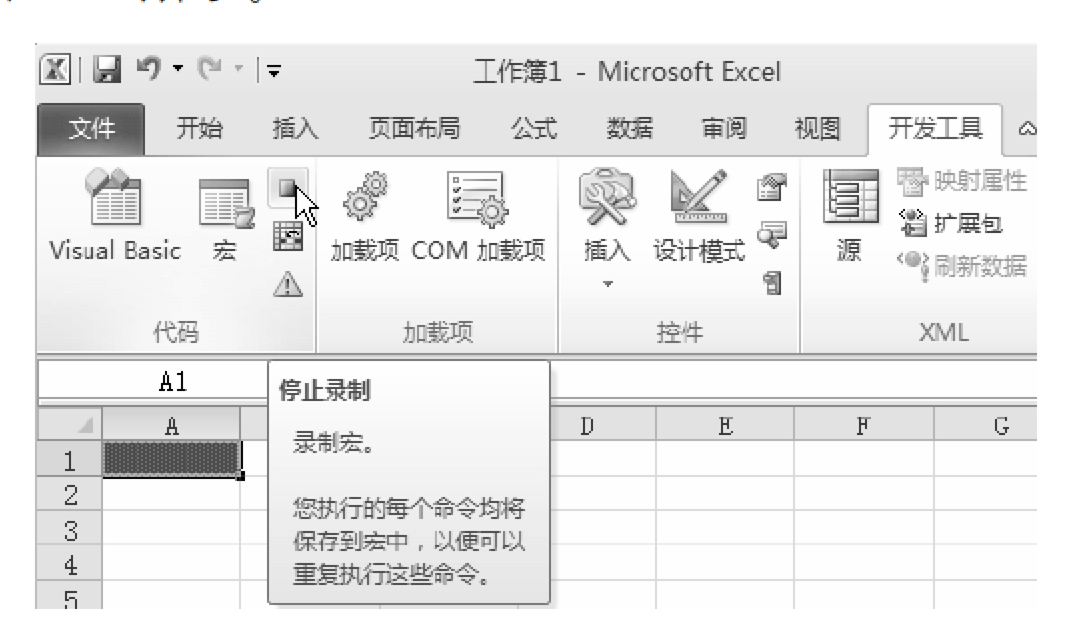


图 3-11　停止录制宏

【提示】如果用户忘记停止宏的录制，系统将会继续录制用户接下来的所有操作，直到关闭工作簿或退出 Excel 应用程序为止。

2. 直接在 VBE 环境中输入宏代码

用户可以直接在 VBE 环境中输入宏代码，具体操作步骤如下。

01 选择【开发工具】选项卡，在【代码】选项组中单击 Visual Basic 按钮，如图 3-12 所示。

02 即可进入 VBE 环境，用户即可快速输入相关宏代码，如图 3-13 所示。

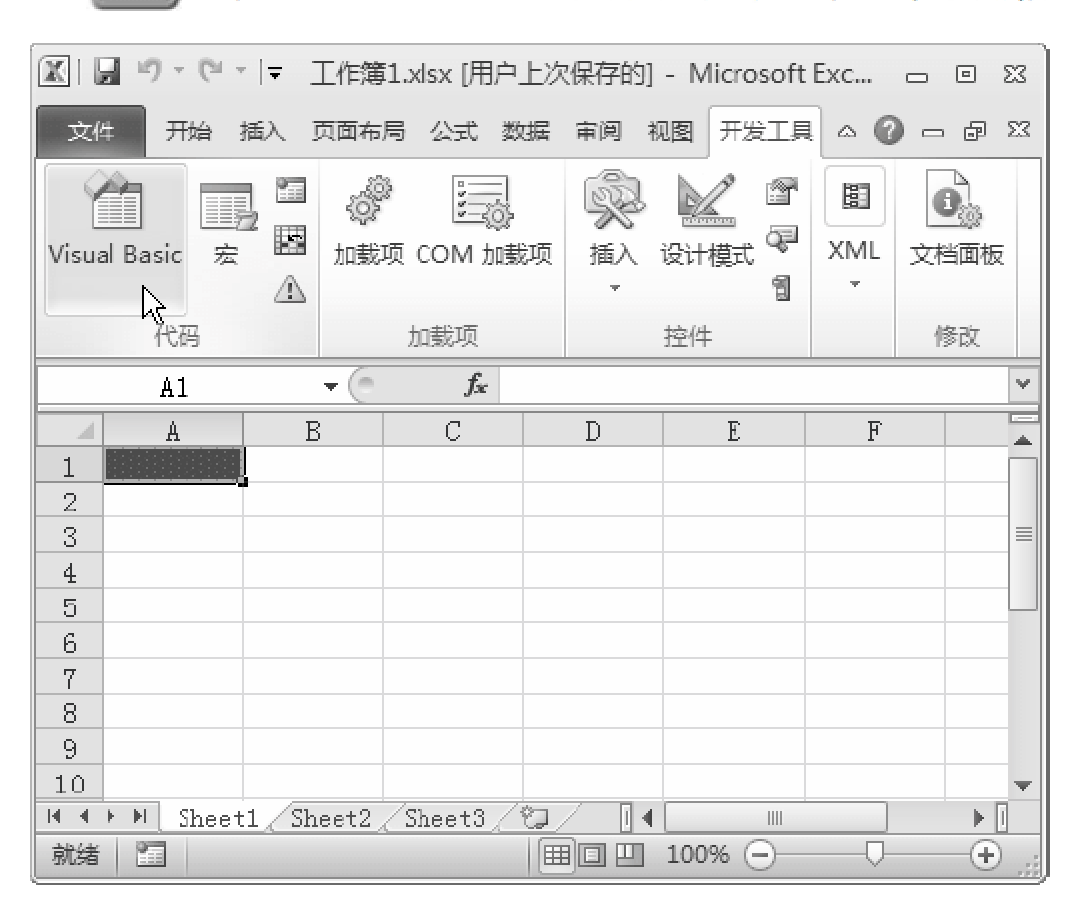


图 3-12　Visual Basic 按钮

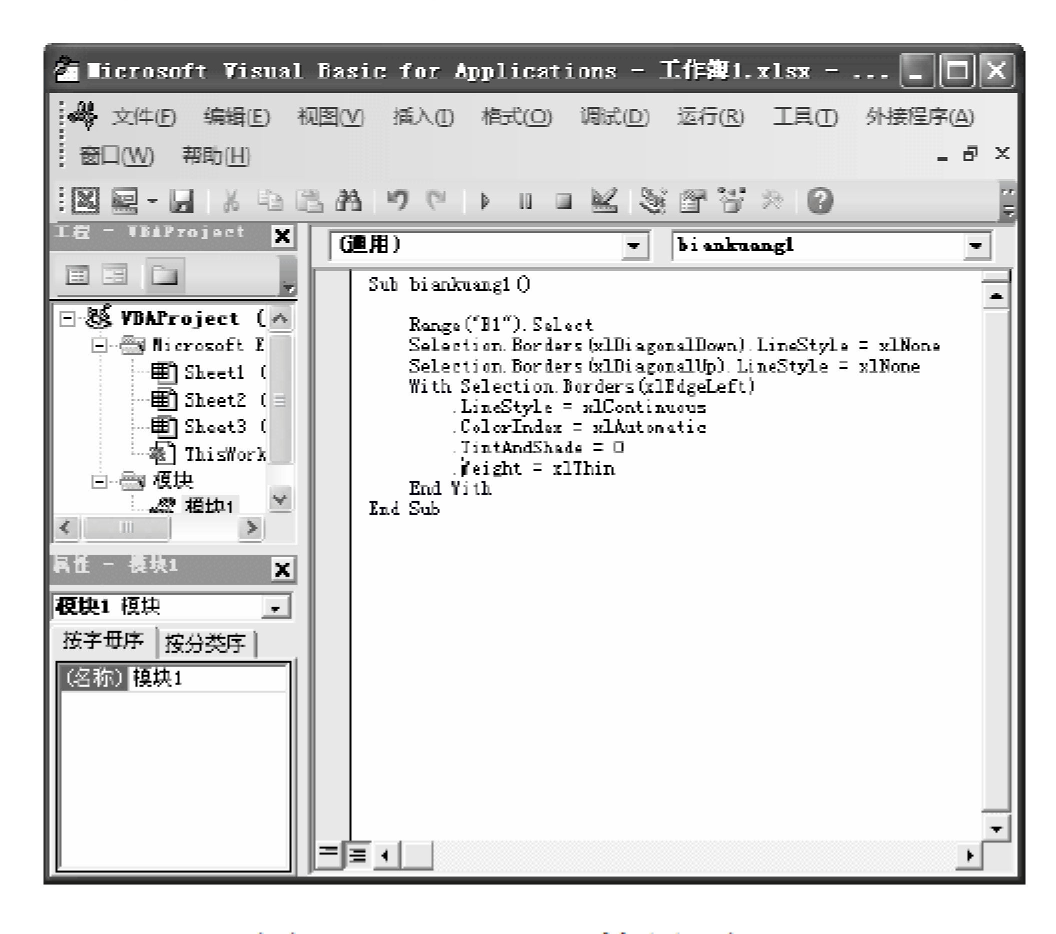


图 3-13　VBE 环境界面

【提示】录制宏只能执行和原来完全相同的操作，而编写的宏可以识别不同的情况以执行不同的操作。可见编写的宏要比录制的宏更能灵活地处理复杂的操作。

3.2.3 查看录制的宏代码

Excel 中的宏录制器非常有用，它不仅能够生成可用的 VBA 代码，而且可以发现相关的对象、方法和属性的名称。虽然由宏生成的代码不是最有效的，但是可以提供很多有用的信息。

学习 VBA 最有效的方式是先录制宏，然后通过查看录制宏的代码和功能相比较，快速学习代码的功能。

查看录制的宏代码的具体操作步骤如下。

01 打开含有录制宏的工作簿，选择【开发工具】选项卡，在【代码】选项组中单击【宏】按钮，如图 3-14 所示。

02 弹出【宏】对话框，选择需要查看代码的宏，单击【编辑】按钮，如图 3-15 所示。



图 3-14　单击【宏】按钮

图 3-15　【宏】对话框

03 进入 VBE 编辑环境，即可查看宏的相关代码，如图 3-16 所示。

04 如果想删除宏，用户可以在图 3-15 所示对话框中单击【删除】按钮，弹出警告对话框，单击【是】按钮，即可删除不需要的宏，如图 3-17 所示。

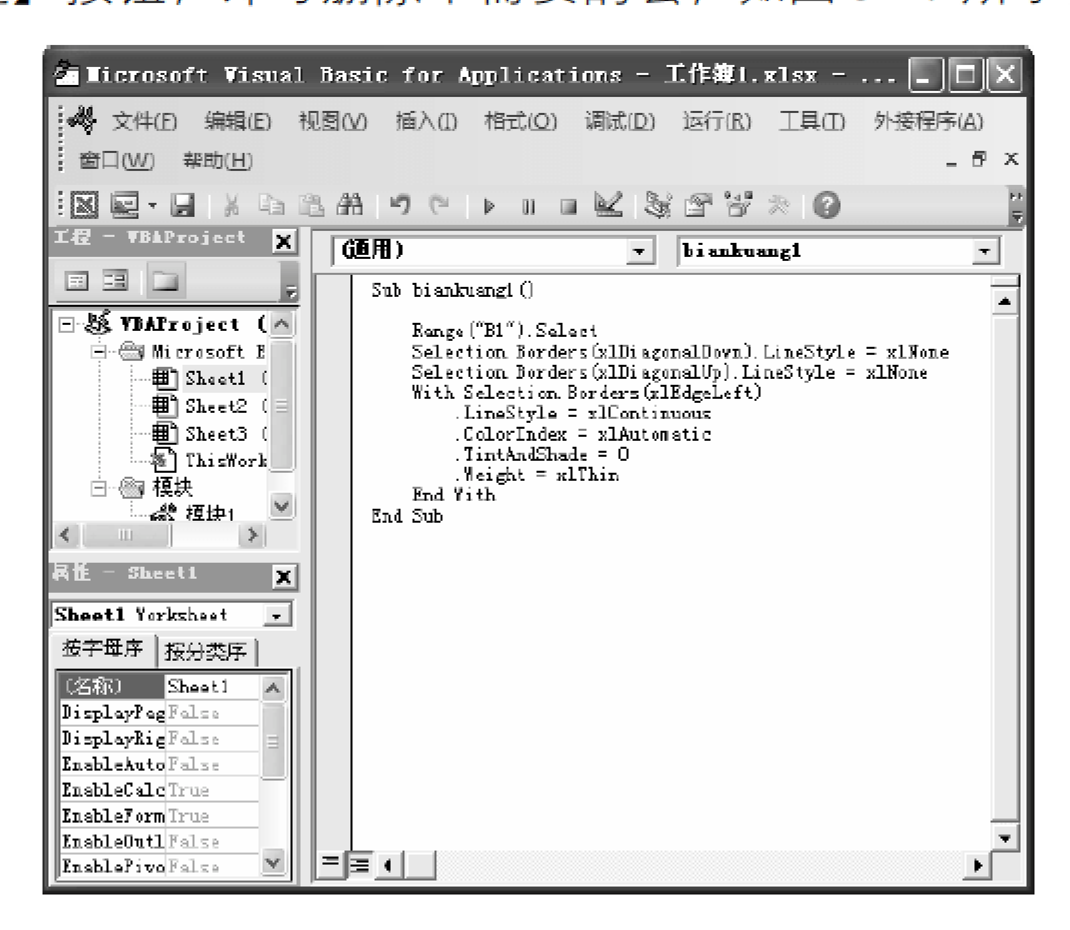


图 3-16　查看宏的代码



图 3-17　警告对话框



3.3 运行宏

宏创建成功后，即可验证宏的正确性。在 Excel 2010 中，用户可以采用多种方法运行宏，本节将讲述这些方法的具体操作步骤。

3.3.1 使用【宏】对话框运行宏

通过【宏】对话框运行宏的具体操作步骤如下。

01 在【开发工具】选项卡的【代码】选项组中单击【宏】按钮，即可打开【宏】对话框，选择需要运行的宏，单击【执行】按钮，如图 3-18 所示。

02 即可看到执行宏后的效果。可以看出此宏的目的是插入一个剪贴画，如图 3-19 所示。



图 3-18 【宏】对话框

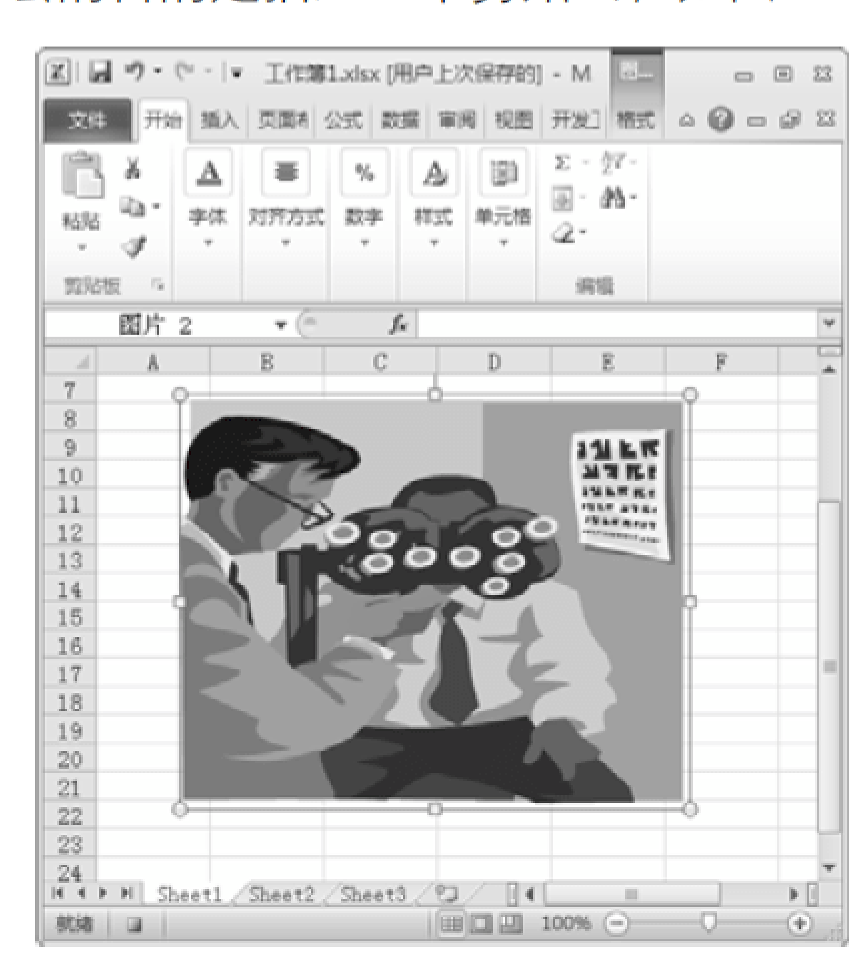


图 3-19 插入剪贴画效果

3.3.2 使用快捷键运行宏

在 Excel 2010 中，用户可以为每一个宏指定一个快捷键，从而提高运行宏的效率，具体操作步骤如下。

01 打开包含宏的工作簿，在【开发工具】选项卡的【代码】选项组中单击【宏】按钮，即可打开【宏】对话框，选择需要添加快捷键的宏，单击【选项】按钮。

02 弹出【宏选项】对话框，在【快捷键】文本框中输入设置快捷键的字母，单击【确定】按钮，如图 3-20 所示。



图 3-20　【宏选项】对话框

3.3.3 使用快速访问工具栏运行宏

对于经常使用的宏，用户可以将其放在快速访问工具栏中，这样可以提高工作效率。具体操作步骤如下。

01 打开包含宏的工作簿，选择【文件】➤【选项】菜单命令，如图 3-20 所示。

02 弹出【Excel 选项】对话框。在【从下列位置选择命令】下拉列表中选择【宏】选项，如图 3-22 所示。



图 3-21　选择【选项】菜单命令



图 3-22　选择【宏】选项

03 选择需要添加的宏名称，例如本实例选择【宏 2】，单击【添加】按钮，然后单击【确定】按钮，如图 3-23 所示。

04 此时在快速访问工具栏上即可看到新添加的【宏 2】按钮，单击此按钮即可运行宏，如图 3-24 所示。





图 3-23　添加宏 2

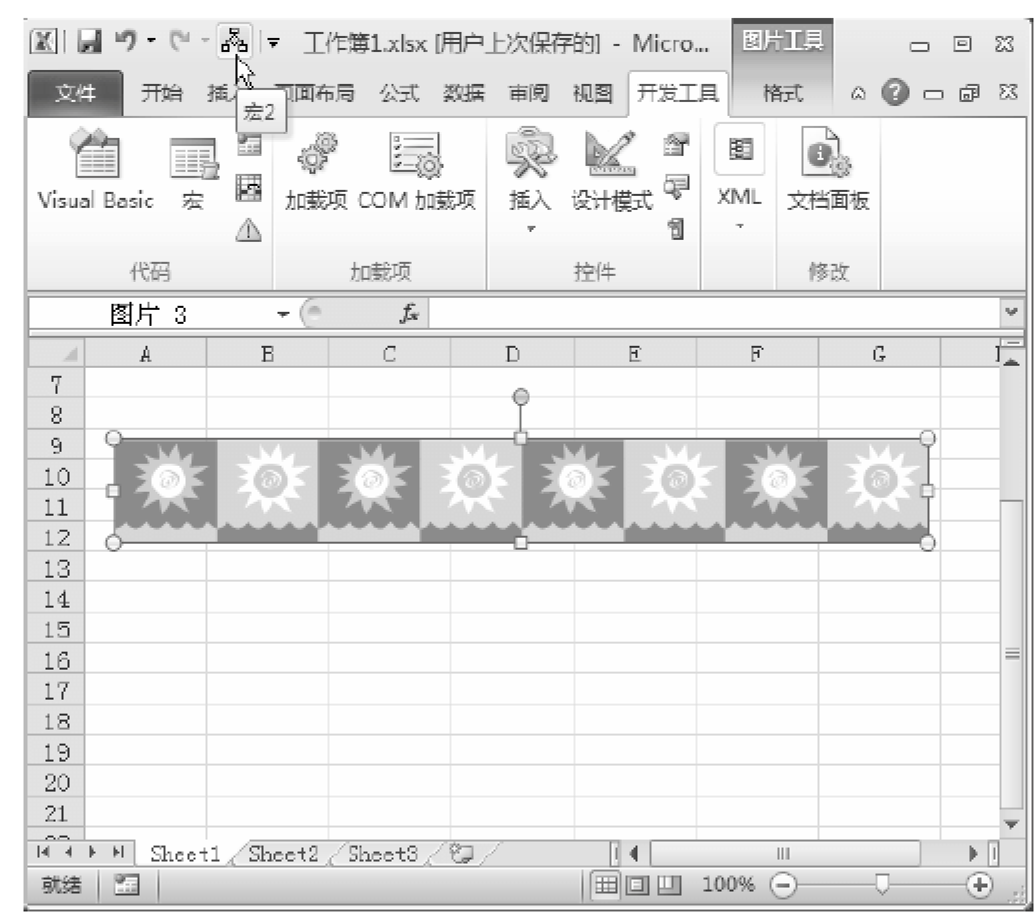


图 3-24　将宏添加到工具栏

3.3.4　保存带宏的工作簿

默认情况下，带宏的工作簿不能保存，此时需要用户自定义加载宏的方法来解决。具体的操作步骤如下。

01 打开含有宏的工作簿，选择【文件】➤【另存为】菜单命令，如图 3-25 所示。

02 即可打开【另存为】对话框，从中选择保存路径后，在【保存类型】下拉列表中选择【Excel 加载宏（*.xlam）】选项，单击【保存】按钮即可，即可加载自定义加载宏文件的过程，如图 3-26 所示。



图 3-25　【另存为】菜单命令



图 3-26　【另存为】对话框



3.4 提高宏的安全性

包含宏的工作簿更容易感染病毒，所以用户需要提高宏的安全性。具体操作步骤如下。

01 打开包含宏的工作簿，选择【文件】➤【选项】菜单命令，打开【Excel 选项】对话框，选择【信任中心】选项，然后单击【信任中心设置】按钮，如图 3-27 所示。

02 弹出【信任中心】对话框，在左侧列表中选择【宏设置】选项，然后在【宏设置】列表中选中【禁用无数字签署的所有宏】，单击【确定】按钮，如图 3-28 所示。



图 3-27　【Excel 选项】对话框

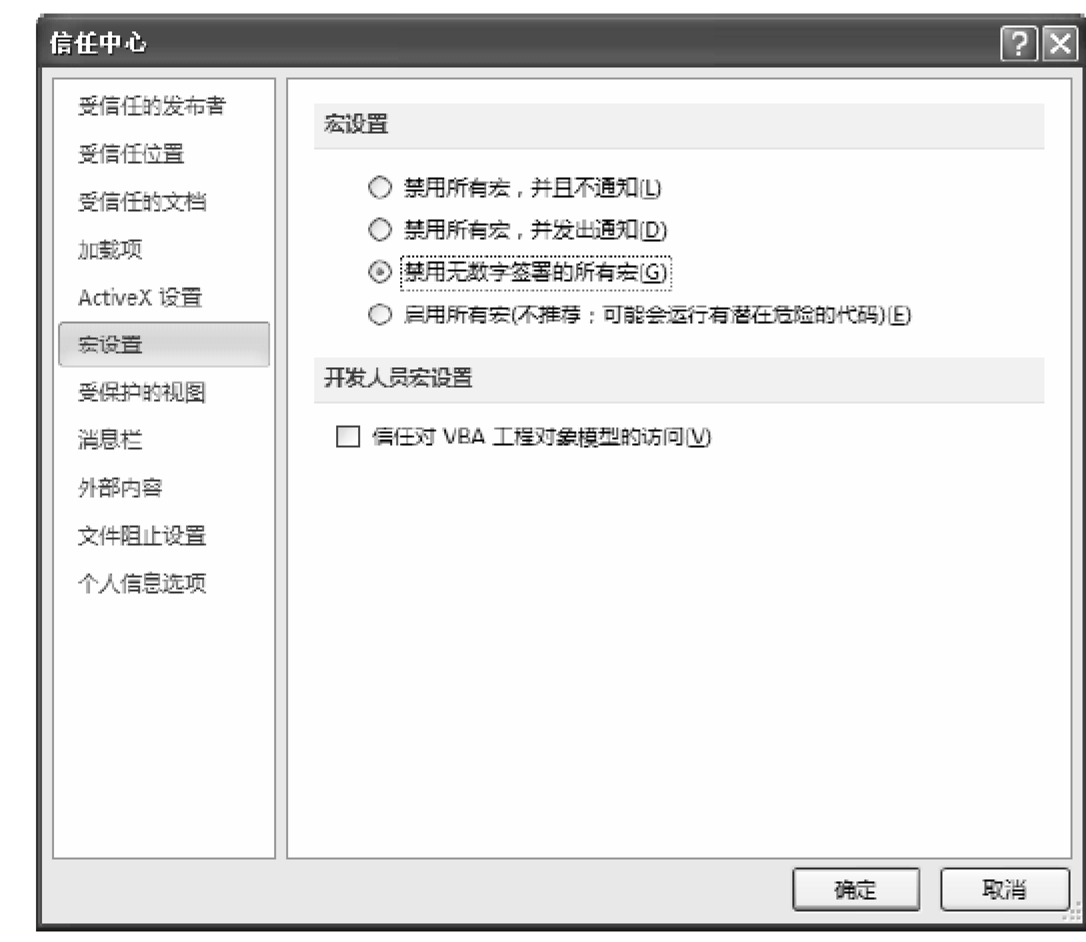


图 3-28　宏的代码

03 单击【开始】按钮，选择【所有程序】➤【Microsoft Office】➤【Microsoft Office 2010 工具】➤【VBA 工程的数字证书】菜单命令，如图 3-29 所示。

04 弹出【创建数字证书】对话框，输入证书的名称，单击【确定】按钮，如图 3-30 所示。



图 3-29　【VBA 工程的数字证书】菜单命令

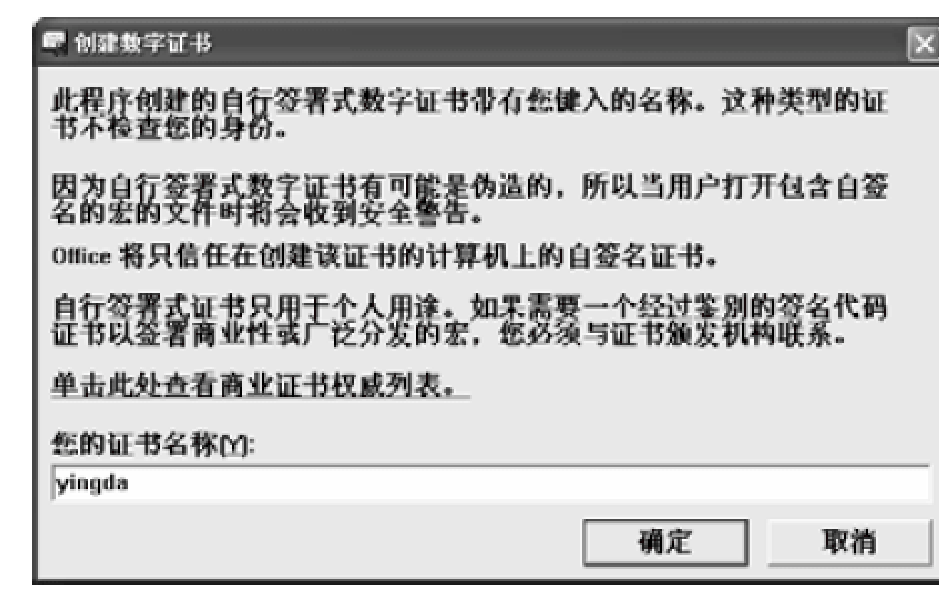


图 3-30　【创建数字证书】对话框

05 弹出提示成功对话框，单击【确定】按钮，如图 3-31 所示。

06 返回到 Excel 2010 主界面，单击【代码】选项组中的 Visual Basic 按钮，如图 3-32 所示。

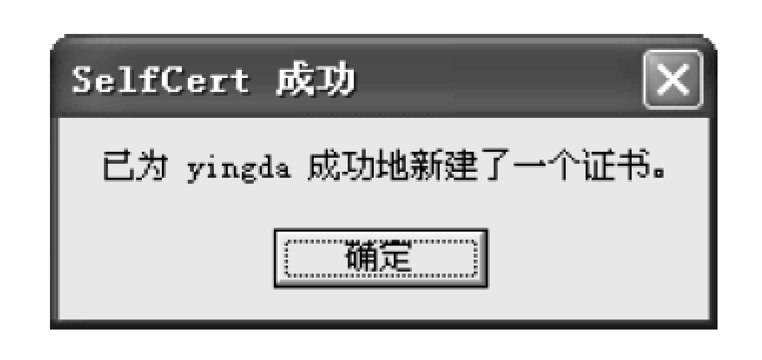


图 3-31 信息提示框



图 3-32 单击【Visual Basic】按钮

07 进入 VBE 环境，选择【工具】➤【数字签名】菜单命令，如图 3-33 所示。

08 弹出【数字签名】对话框，单击【选择】按钮，如图 3-34 所示。



图 3-33 【数字签名】菜单命令



图 3-34 【数字签名】对话框

09 弹出【选择证书】对话框，选择上面创建的的证书，单击【确定】按钮，如图 3-35 所示。

10 返回到【数字签名】对话框，单击【确定】按钮成功添加数字签名，如图 3-36 所示。保存该 Excel 文件，完成数字签名操作。




图 3-35 【选择证书】对话框



图 3-36 选择数字签名后效果



3.5 综合实战——录制自动排序的宏

在实际工作中，只需把在 Excel 工作表内的操作过程录制下来，便可以解决一些重复性的工作，大大提高工作效率。

本实例主要录制自动排序的宏，具体操作步骤如下。

01 新建销售报表并输入以下内容，然后选择【开发工具】选项卡，在【代码】选项组中单击【录制宏】按钮，如图 3-37 所示。

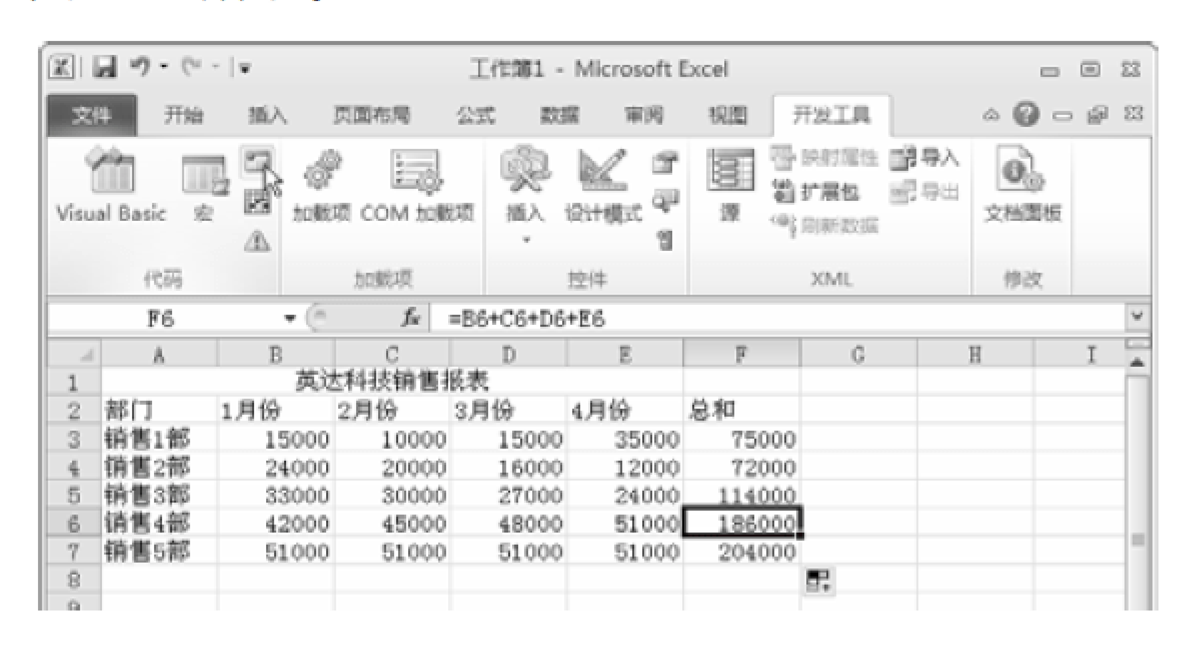


图 3-37　单击【录制宏】按钮

02 弹出【录制新宏】对话框，在【宏名】文本框中输入“数据排列”，单击【确定】按钮，如图 3-38 所示。



图 3-38　宏的代码

03 选择 A2：F7 单元格区域，然后选择【数据】选项卡，在【排序和筛选】选项组中单击【排序】按钮，如图 3-39 所示。

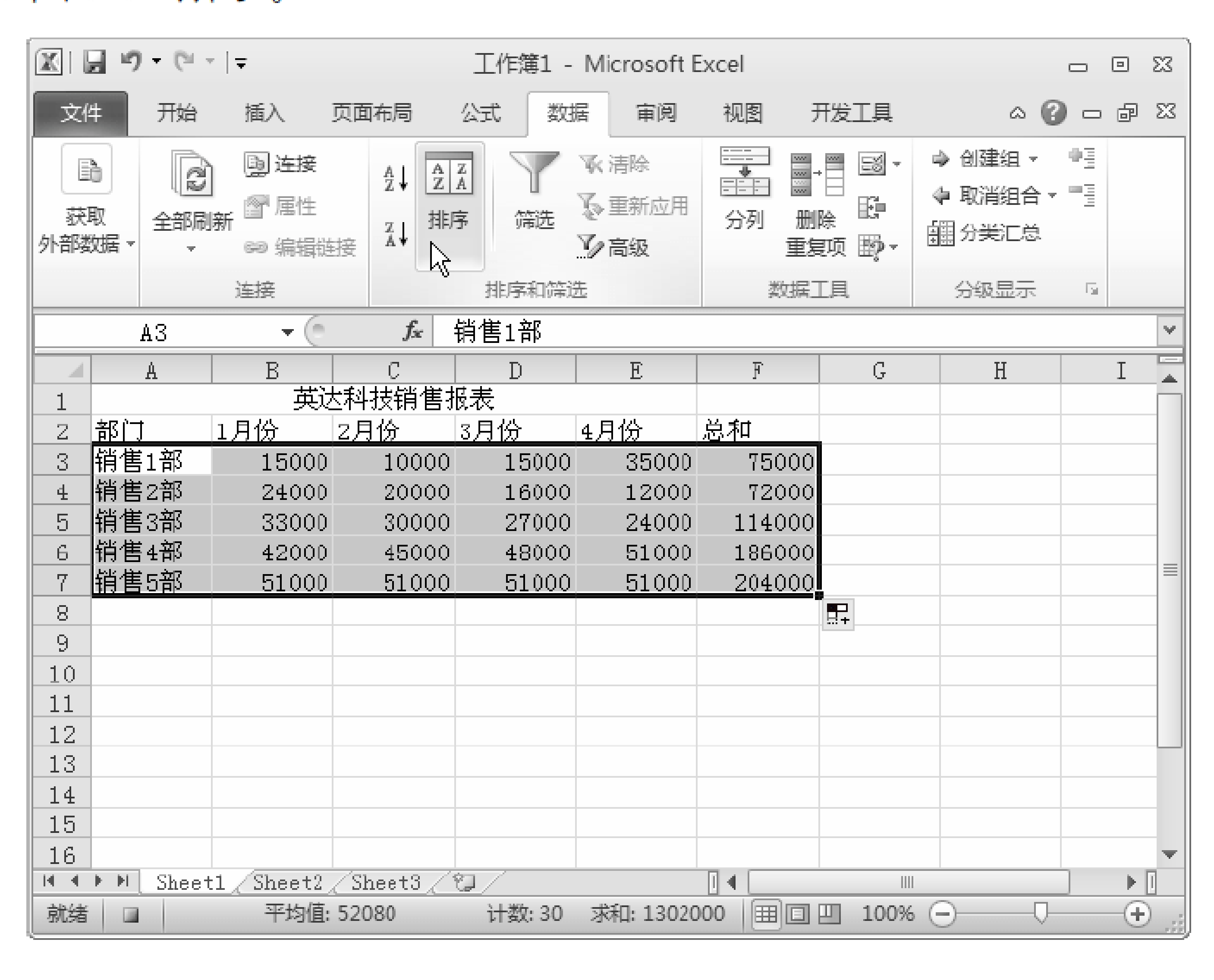


图 3-39　单击【排序】按钮

04 弹出【排序】对话框，选择【主要关键字】为【总和】选项，然后单击【添加条件】按钮，选择【次要关键字】为【1 月份】，单击【确定】按钮，如图 3-40 所示。



图 3-40　【排序】对话框

05 单击【代码】选项组中的【停止录制】按钮，即可完成数据排序宏的录制，如图 3-41 所示。

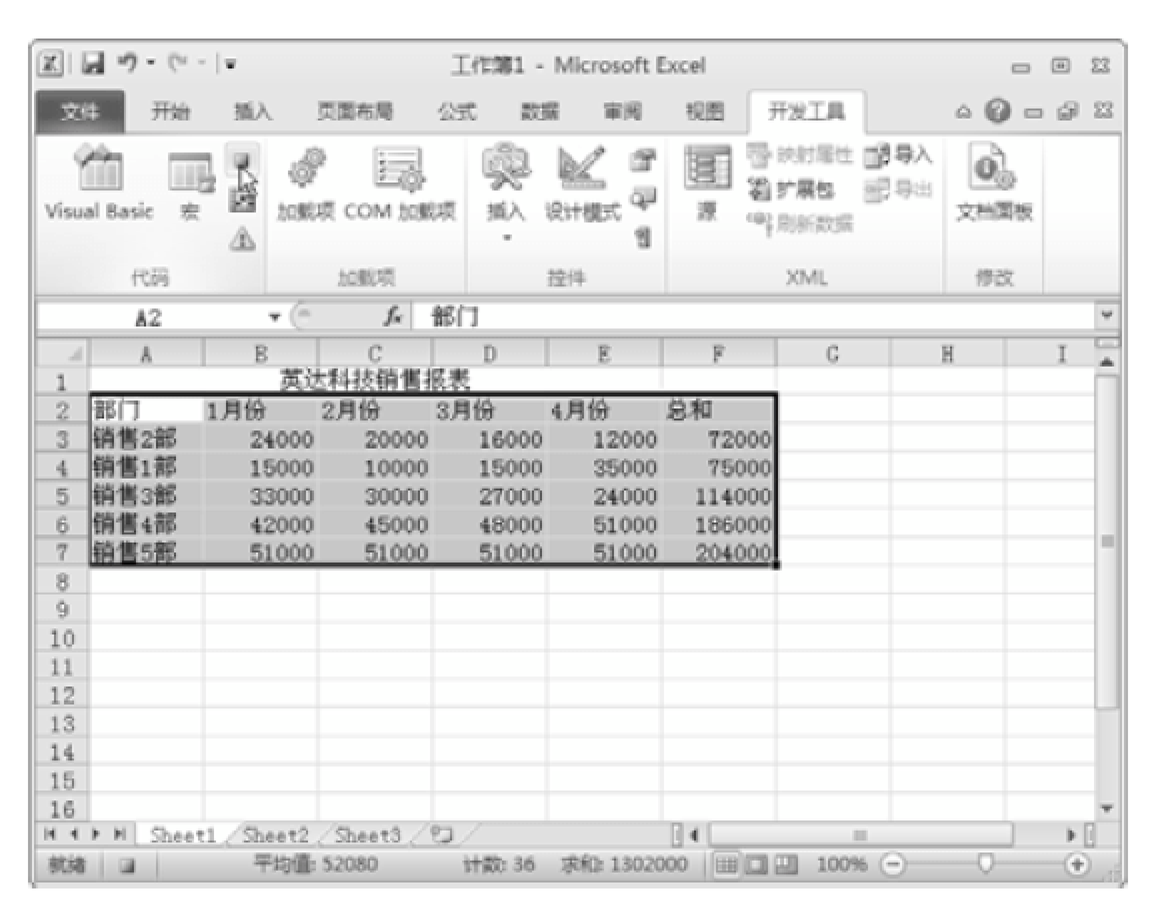


图 3-41　停止录制宏



3.6 高手私房菜

技巧 1：宏自动启动有秘诀

默认情况下，宏需要用户手动启动。录制宏时，在【录制新宏】对话框中可以将宏命名为“Auto_Open”，即可在工作簿运行时自动启动宏，如图 3-42 所示。另外对于创建好的宏，可以在 VBE 环境中直接修改宏名称为“Auto_Open”即可，如图 3-43 所示。



图 3-42　【录制新宏】对话框

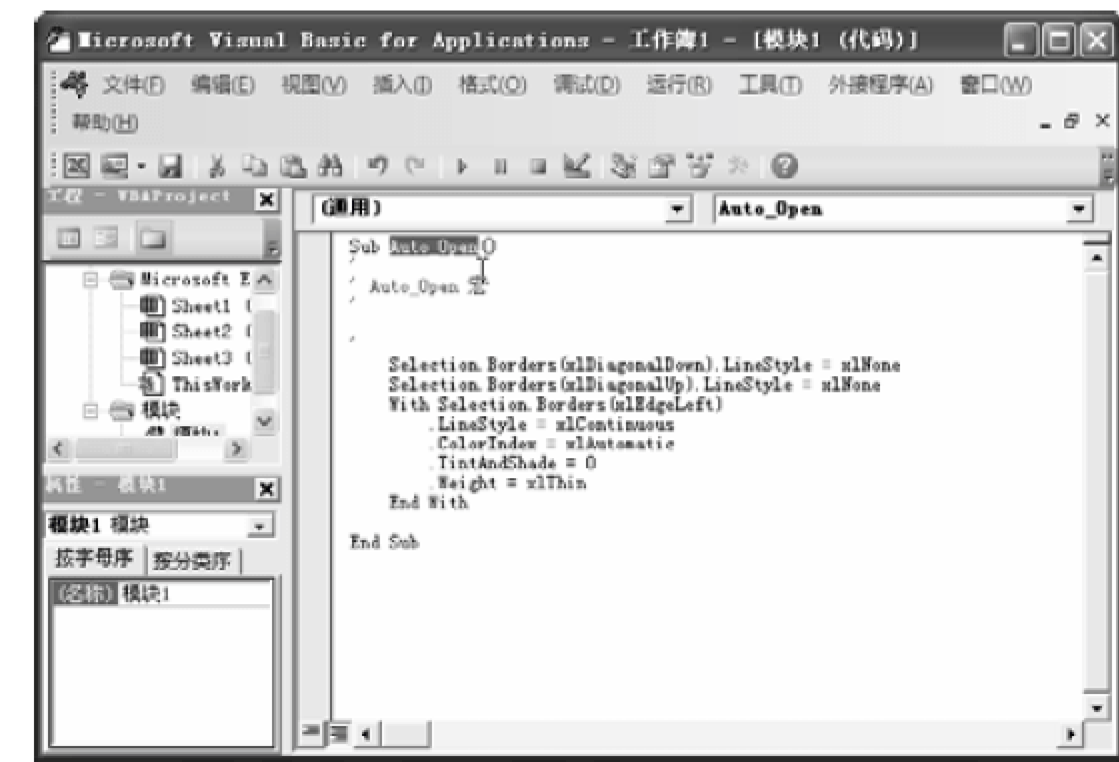


图 3-43　修改宏名称

技巧 2：宏出现错误时的处理妙方

若正在运行中的宏出现错误，则指定的方法不能用于指定的对象，其中原因很多，包括参数包含无效值、方法不能在实际环境中应用、外部链接文件发生错误和安全设置等。

其中前 3 种问题，用户可以根据提示检查代码和文件即可，对于安全设置问题比较常见，用户可以单击【开发工具】选项卡【宏】选项组中的【宏安全性】按钮，在弹出的【信任中心】对话框中选中【信任对 VBA 工程对象模型的访问（V）】复选框，然后单击【确定】按钮即可，如图 3-44 所示。

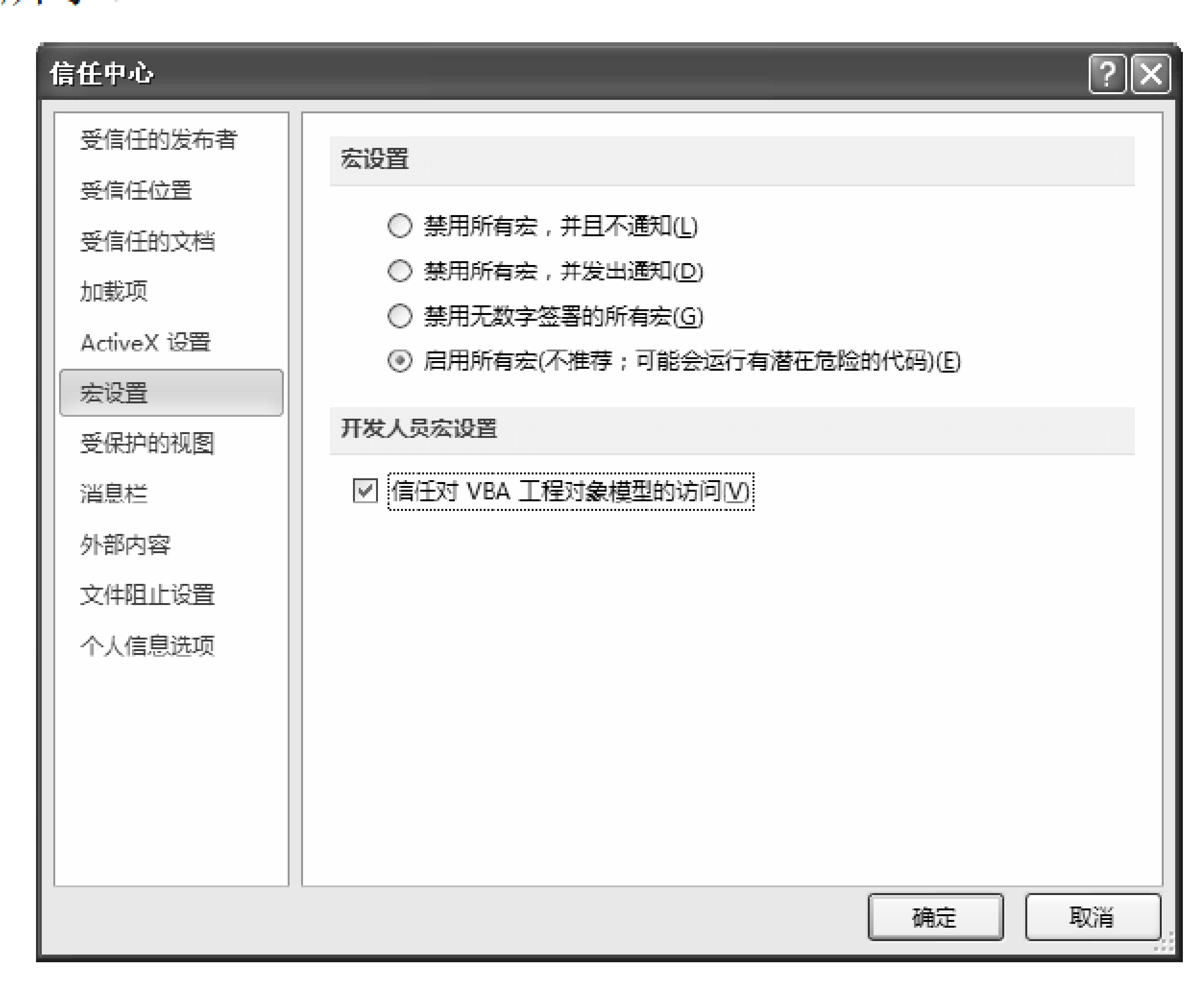


图 3-44　【信任中心】对话框

第 4 章　VBA 基础

在使用 VBA 进行编程前，用户首先要学习 VBA 的基础知识，包括 VBA 的概述、常见的数据类型、常量、变量、运算符和表达式等相关知识。通过本章节的学习，用户将熟练掌握 VBA 编程的基本语法知识。



4.1 VBA 简介

VBA 是 Visual Basic for Applications 的缩写，是 Visual Basic 的一种宏语言，主要是用来扩展 Windows 的应用程式功能。例如本书讲述的 Microsoft Excel 2010 软件，通过编写程序能使 Excel 自动完成工作，从而实现应用程序自动化。

VBA 是基于 Visual Basic 发展而来的，它们具有相似的语言结构。Visual Basic 是 Microsoft 的主要图形界面开发工具。



4.2 VBA 的数据类型

数据类型是按照被定义变量的性质、表示形式和构造特点来划分的不同的操作对象。例如要表示一个员工的信息，需要保存员工的姓名、年龄和性别等，这些数据在 VBA 中分别需要使用不同的数据类型来表示。因此，在学习如何使用 VBA 进行程序设计之前，有必要先了解数据类型的相关知识。VBA 中包含的主要数据类型有：布尔型、日期型、字符串型、货币型、小数型、字节型和整型。

表 4-1　VBA 的数据类型

数据类型	关键字	占用空间	取值范围
布尔型	Boolean	2 字节	True 或 False
日期型	Date	8 字节	1/1/100~12/31/9999
字符串型	String	字符串长	
货币型	Currency	8 字节	

（续表）

数据类型	关键字	占用空间	取值范围
单精度型	Single	4 字节	负数：－3.402823E38~－1.401298E-45 正数：1.401298E－45~3.402823E38
双精度型	Double	8 字节	负数：－1.79769313486231E308~－4.94065645841247E－324 正数：4.94065645841247E－324~1.79769313486231E308
小数型	Decimal	14 字节	
字节型	Byte	1 字节	0~255
整型	Integer	2 字节	-32768~32767
长整型	long	4 字节	-2147483648～2147483647

下面将详细介绍各种数据类型的特点。

4.2.1 布尔型

布尔型（Boolean）用来表示逻辑的真值（True）和假值（False）。布尔型变量的显示值为 True 或 False，存储时占用 2 字节。定义完布尔类型的变量之后，可以使用 True 或 False 为这些变量赋值。

布尔类型值可以与其他类型值进行转换，当其他类型的数值转换为布尔值时，0 值转换为 False，其他值变成 True；当将布尔值转换为其他的数据类型时，False 转换为 0，而 True 转换为-1。

4.2.2 日期型

日期型（Date）数据用来存储日期和时间。使用日期型数据时需要采用“#”符号把日期和时间值括起来，例如#2011-01-01#，#06/01/2010 10:05 AM#。

4.2.3 字符串型

字符串型（String）用来存储字符序列。字符串数据在使用时需要包括在双引号内，字符串中也可以不包括任何字符，这样的字符串叫做空字符串。下面是一些字符串的表示形式：

```
"欢迎大家学习 VBA"
"Today is Monday!"
"1008611"
""
```

上面 4 个数据都是字符串类型的，最后一个形式表示一个空字符串。

VBA 中的字符串可以分为两种：定长字符串与变长字符串。定长字符串是长度固定的字符串，在定义时指定字符串中可以保存的字符的长度是固定的；变长字符串是长度可变的字符串，其实际字符串的长度由其中包含的实际字符个数决定。

4.2.4 货币型

货币型（Currency）数据专门用来处理货币。货币类型数据在存储时占用 8 字节，货币数据类型在声明时使用“@”符号，货币数据类型主要适用于对精度要求比较高的场合，其可以表示的存储范围为-922 337 203 685 477.580 8~922 337 203 685 477.580 7。

4.2.5 单精度和双精度型

单精度（Single）和双精度（Double）都是可以用来表示带有小数的数据。单精度和双精度以指数的形式来表示数。

Single 表示分数、带小数位或指数的数值等单精度数，占用 4 字节的存储空间，取值范围为负值时-3.402 823E38~-1.401 298E-45， 正值时 1.401 298E-45~3.402 823E38。

Double 存储双精度浮点数，占用 8 字节的存储空间，取值范围为负值时-1.797 693 134 862 32E308~-4.940 656 458 412 47E-324，正值时 1.797 693 134 862 32E308~4.940 656 458 412 47E-324，默认值为 0。

4.2.6 小数型

小数型（Decimal）是一种包含以 10 的幂为刻度的十进制数的变体子类型，不是一种独立的数据类型。占用 14 字节的存储空间，取值范围为±79 228 162 514 264 337 593 543 950 335（不带小数点）或±7.922 816 251 426 433 759 354 395 033 5（带 28 位小数点），默认值为 0。

4.2.7 字节型

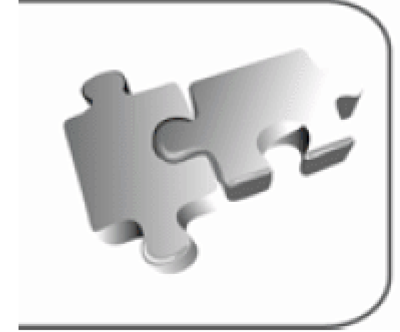
字节型（Byte）用来存储二进制数，以正整数值的形式保存，存储范围为 0~255，占用 1 字节的存储空间。

4.2.8 整型

整型用来存储不带小数部分的数，如正整数、负整数和零。整型分为普通整型（Integer）和长整型（Long）。

Integer 表示-32768~32767 之间的整数，其中一位表示符号，占用 2 字节的存储空间，默认值为 0。

Long 表示-2 147 483 648~2 147 483 647 之间的整数，其中一位表示符号，占用 4 字节的存储空间，默认值为 0。



4.3 常量和变量

常量是在程序执行之前已经确定的值，在程序执行时，其中保存的值或字符串不会发生改变。

4.3.1 常量的分类

VBA 有三种形式的常量：直接常量、符号常量和系统常量。用户可以指定直接常量和符号常量的值，而系统常量则是 VBA 提供的一些系统内部已经定义好的常量，可以直接使用。

1. 直接常量

直接常量也叫一般常量，其中保存用户直接书写的值，例如 4.2.3 节中介绍的几个字符串数据都是直接常量，也可以指定数值常量或日期常量等，如 18，#2012-2-20#。

2. 符号常量

符号常量是在程序中使用的用符号表示的常量，使用符号常量可以减少程序出错的可能性、加快程序的修改和提高程序的可读性。定义符号常量的语法格式如下。

```
Const 常量名称 [AS 数据类型]=常量值表达式
```

这里 Const 是定义符号常量的关键字，等号后面常量值表达式计算后的结果将保存在常量名称中，保存之后，用户就不能修改常量名称中保存的值了。例如：

```
Const VAR1=365
Const MSG= "Happy birthday!"
```

上面分别声明了一个整型常量 VAR1 和字符串常量 MSG。使用 Const 也可以一次定义多个常量，例如：

```
Const NAME="李阳", PI AS Double=3.1415;
```

3. 系统常量

系统常量根据所在系统库可以分为 VBA 系统常量和 Excel 系统常量。

系统常数采用大小写混合的格式，其前缀表示定义常数的对象库名称。VBA 系统内部符号常量以小写字母“vb”开头，例如色彩常量“vbBlack”表示黑色、“vbSunday”表示周日；Excel 内部系统常量通常以小写字母“xl”开头。用户可以在“对象浏览器”中查看这些系统变量的具体值，如图 4-1 所示。





图 4-1 对象浏览器

常量在命名时一般都使用大写字母，以与一般的变量区别开来。

4.3.2 变量

变量是在程序运行过程中可以改变值的量，变量提供了一个存储数据的容器，变量在使用之前应当先要声明，之后系统将会给变量分配一个存储单元，变量声明时要指定变量名称和数据类型，不同类型的变量中保存的数据也是不同的，程序中通过变量的名称引用其中的值。变量的名称在其作用域内必须是唯一的。根据变量的作用域不同，可分为局部变量、全局变量。

变量命名中可以指定任何英文或者中文字符，但声明变量时要注意以下几点：

（1）变量必须以字母开头，中间可以出现数字和一些标点符号，除下划线（_）作为连字符外，变量名称不能有空格、加号（+）、减号（-）、逗号（,）、小数点（.）等符号。

（2）VBA 不区分大小写。但在变量命名时，最好体现该变量的作用。

（3）不能使用 VBA 中的关键字作为变量。

（4）变量名称中不能有特殊类型的声明字符（#、$、%、&或!）。

（5）变量名称最多可以包含 254 个字符。

4.3.3 定义变量

变量使用之前需要声明，声明变量使用 Dim 语句，虽然在 VBA 中可以通过不声明而直接在程序中使用变量，这样系统会自动创建一个变量，但通常情况下，代码编写人员应该养成良好的编程习惯，在使用之前强制声明变量，这样 VBA 也可以知道变量的名称和数据类型。Dim 语句的基本语法格式如下：

```
Dim 变量名称 [AS 数据类型]
```

Dim 为声明变量的关键字。

AS 关键字后面指定变量的数据类型，即其中要保存什么数据类型的数据，例如 String、Integer、Date 等。

这里“AS 数据类型”使用中括号括起来，表示在使用 Dim 声明变量时可以不指定 AS 关键字后面的数据类型，系统会根据指定的值自动为改变量指定数据类型。

变量的声明可以放在程序的开始处或者是代码中间，一般情况将其放在程序的开始处。例如，下面是一些声明变量的语句：

```
Dim strStudentName = "楚飞"
Dim discount AS Single
Dim birthday AS Date
```

上面分别声明了 3 个变量，第一个是 String 类型的变量，第二个是单精度的小数，第三个是日期类型的变量。

另外，多个变量也可以在同一个 Dim 语句中声明，此时需要制定每一个变量的数据类型。例如：

```
Dim intA AS Integer, intB AS Integer, intC AS Integer
```

该语句声明了 3 个 Integer 类型的变量，名称分别为 intA、intB 和 intC。

接下来，看下面的一条 Dim 语句：

```
Dim intA intB, intC AS Integer
```

intA、intB 和 intC 的数据类型还是 Integer 吗？不是，因为 VBA 中变量在没有指定数据类型时，默认都为 Variant 类型，所以这里 intA 和 intB 的数据类型是 Variant，而只有 intC 的数据类型是整型。读者在声明变量时一定要注意为每个变量指定数据类型。

4.3.4 变量的定义域

变量在不同的地方定义时，有不同的使用范围，根据变量的使用范围，可以将变量分为过程级变量、模块级变量和全局变量。

1. 过程级变量

在过程中声明的变量只能在过程中使用，即变量定义域是过程级的，这种变量也称为局部变量。在声明局部变量的过程结束的同时，变量也失去作用，并且系统释放变量占用的内存空间。所以可以在不同的过程中声明具有相同名称的变量。

2. 模块级变量

变量如果在模块中声明，则在该模块的每个过程中都可以使用，即变量的定义域是模块级的。这样可以实现多个过程之间对某个变量的共享。定义模块级的变量时，Dim 语句必须

放在模块的声明部分，一般情况是在 Option Explicit 语句和第一个模块之间。例如：

```
Option Explicit
Dim stuScore AS Integer
Sub func1()
     stuScore =50
    MsgBox stuScore
End Sub
Sub func2()
     stuScore =stuScore + 20
    MsgBox stuScore
End Sub
```

上面的两个代码定义了两个模块，即函数 func1 和函数 func2，这两个函数都可以访问 stuScore 变量，这里 Dim 关键字也可以换成 Private 关键字。

3. 全局变量

全局变量是在应用程序的所有模块中或窗体中都可以访问的变量。全局变量在模块的通用声明段使用 Public 语句声明，例如：

```
Option Explicit
Public intCount AS Integer
```

这里声明的计数变量 intCount 可以被程序中的窗体、模块直接引用。

【提示】程序中可能会有许多变量，一般情况下，应该尽量使用局部变量，只有要在不同过程之间共享数据时，才使用模块变量或者是全局变量。

4.3.5 变量的生存期

变量的定义域表示其在空间上的作用范围，而生存期是表示变量在时间上的范围，即变量在内存中保留的时间。变量的作用域决定了该变量内存中保留的时间。

对于过程级或模块级的变量，每次调用过程或模块时，对过程中的变量初始化，退出时将释放变量所占的内存，变量随之消失。

如果将过程级或模块级的变量声明为静态变量，则在退出过程或模块时，变量中的值仍将保留在内存中，下次调用时可以继续访问。Static 关键字可以用来声明一个或多个静态变量，其语法格式如下：

```
Static 变量名称 AS 数据类型
```

【例 4.1】比较静态变量和普通变量的运行结果，操作过程如下。

01 进入 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 4-1 所示。

```
Sub testStatic1()
```

```
    Dim Count1 As Integer
    Static s Count As Integer
    s_Count = s_Count + 1
    Count1 = Count1 + 1
    Debug.Print "Count1 :"; Count1
    Debug.Print "s_Count : "; s_Count
End Sub
```

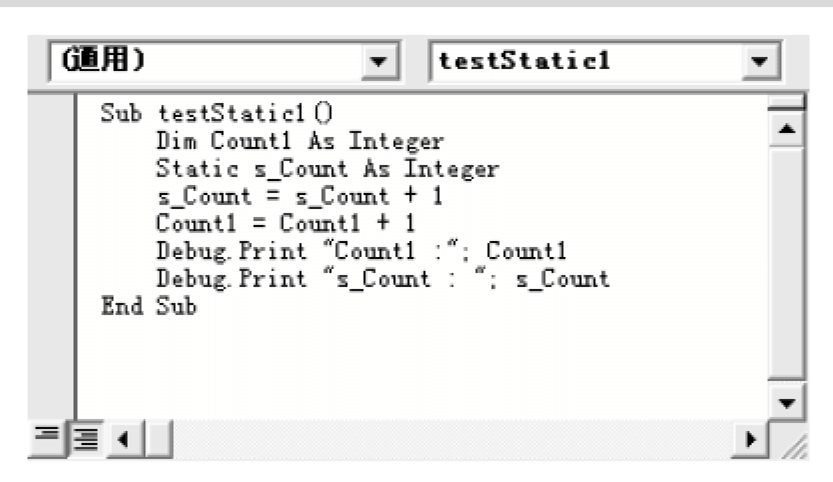


图 4-1　testStatic1 过程代码

03 连续单击工具栏中的【运行子过程/用户窗体】按钮（带绿色小三角图标的按钮）3 次，使该子过程运行 3 次，结果输出在【立即窗口】中。

如图 4-2 所示，程序中的 Count1 是一个局部变量，该过程每次执行时 Count1 都会被初始化，执行完毕之后将其自动释放，所以 Count1 的值一直都是 1。

而变量 s_Count 被定义为静态变量，只有第一次进入模块时才进行初始化操作，但是当退出时 s_Count 并没有被释放，再次执行该模块时，s_Count 将在保留此值基础上再次加 1，所以每次执行一次模块，变量 s_Count 值就增加 1，结果如图 4-2 所示。

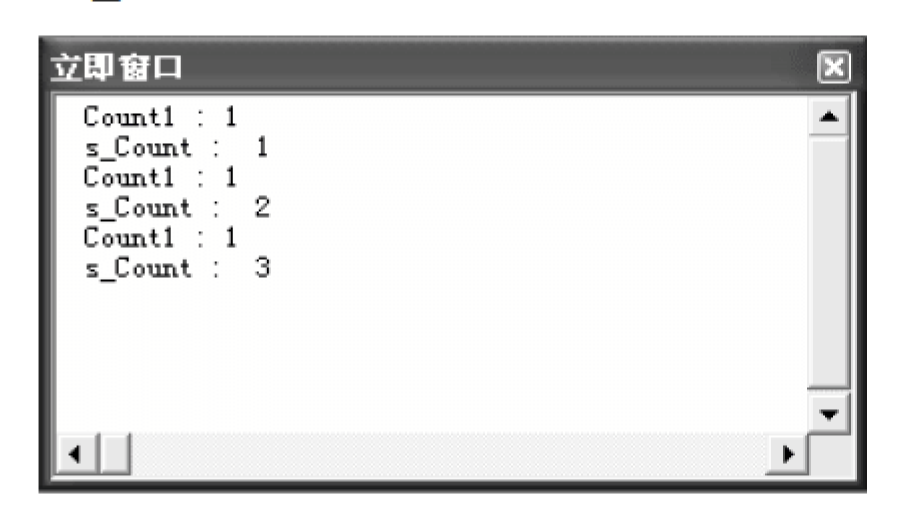


图 4-2　静态变量的生存期



4.4 表达式和运算符

运算符是对特定数据执行运算任务的操作符，将常量、变量和函数等运用运算符连接起来的运算式称为表达式。常见的有 4 种基本的运算符：算数运算符、比较运算符、逻辑运算符和连接运算符。本小节将分别介绍这几类运算符。

4.4.1 算术表达式

算术表达式是由算术运算符与操作数构成。算术运算是计算机的基本功能，VBA 中主要有如下几个算术运算符。

- +：加号运算符，进行加法运算，如 3+3 结果为 6。
- －：减法运算符或取负运算符，进行减法运算 3－1 结果为 2，-2 表示一个负数。
- *：乘法运算符，进行乘法运算，如 3*3 结果为 9。
- /：浮点除法运算符，如除法运算 10/3 结果为 3.333 333 3。
- \：整数除法运算符，如除法运算 10\3 结果为 3。
- Mod：取模运算符，返回两个数的除法运算后的余数，如 12 Mod 5 余数为 2。
- ^：指数运算符，进行乘幂运算，如 3^2 结果为 9。

4.4.2 比较表达式

比较表达式是由比较运算符与操作数构成。比较运算符用于比较两个或多个值或者表达式之间的关系。比较运算符的结果是一个逻辑值，不是 TRUE 就是 FALSE。VBA 中有如下几个比较运算符。

- =：等于运算符，用于比较两个值是否相等，如 1=1 结果为 TRUE。
- >：大于运算符，用于比较运算符左边的值是否大于右边的值，如 1>2 结果为 FALSE，3>1 结果为 TRUE。
- <：小于运算符，用于比较运算符左边的值是否小于右边的值，如 1<2 结果为 TRUE，3<1 结果为 FALSE。
- >=：大于等于运算符，用于比较运算符左边的数是否大于或者等于右边的数，如 1>=2 结果为 FALSE，2>=2 结果为 TRUE，3>=1 结果为 TRUE。
- <=：小于等于运算符，用于比较运算符左边的数是否小于或者等于右边的数，如 1<=2 结果为 TRUE，2<=2 结果为 TRUE，3<=1 结果为 FALSE。
- <>：不等于运算符，用于两个数的不相等判断，如 1<>2 结果为 TRUE，1<>1 结果为 FALSE。

4.4.3 逻辑表达式

逻辑表达式是由逻辑运算符和操作数构成，逻辑运算符用来对操作数执行逻辑运算，其运算结果只有 TRUE 和 FALSE 两种。VBA 中常用的逻辑运算符如下：

- Not：用来对操作数进行逻辑否运算。
- And：用来对两个操作数进行逻辑与运算。
- Or：用来对两个操作数进行逻辑或运算。
- Xor：用来对两个操作数进行逻辑异或运算。

- Eqv：用来对两个操作数进行逻辑等价运算。
- Imp：用来对两个操作数进行逻辑蕴涵运算。

假如 A 和 B 分别代表两个任意的操作数，T 代表逻辑真值（TRUE），F 代表逻辑假值（FALSE），则各种逻辑运算结果如表 4-2 所示。

表 4-2　逻辑运算符真值表

操作数 A	F	F	T	T
操作数 B	F	T	F	T
NOT A	T	T	F	F
A And B	F	F	F	T
A Or B	F	T	T	T
A Xor B	F	T	T	f
A Eqv B	T	F	F	T
A Imp B	T	T	F	T

4.4.4　连接表达式

连接表达式是由连接运算符与操作数构成。连接运算符用来连接两个以上的字符串，使其成为一个单一的字符串，VBA 中有两个连接运算符，分别是&运算符和+运算符。

1. &运算符

&运算符可以把多个字符串连接在一起，形成一个新的字符串，任何非字符串变量或表达式在进行连接运算之前都会被转换为字符串，例如：

```
 “VB FOR” & “Application”的结果为”VB FOR Application”；
 “Excel ” & 2010 的结果为”Excel 2010”
4&5 的结果是字符串”45”。
```

2. +运算符

+运算符既可以做加法运算，也可用于字符串运算，将多个字符串连接起来，形成一个新字符串，但是与&运算符不同，+运算符只有在操作数都是字符串时才进行连接运算，例如：

```
 “VB FOR” + “Application”结果为”VB FOR Application”
 “” + 2010 运算结果出错，提示“类型不匹配”的错误消息。
4 + 5 的结果为整数 9。
```





4.5 高手私房菜

技巧 1：快速相互转换不同类型的数据

VBA 中提供了大量的数据类型，这些不同的数据类型一般情况下应用在不同的场合，但在某些特殊条件下，用户通过使用 VBA 提供的各种转换函数就可以实现不同数据类型的相互转换。例如：CBool（expression）函数将返回表达式的值是一个已转换为 Boolean 子类型的 Variant 变量。这里 expression 是任意有效的表达式。如果 expression 是 0，则返回 False；否则返回 True。

```
Dim C, D, Check
C = 5: D = 5           ' 初始化变量
Check = CBool(C = D)  ' check 的值为 True
C = 0                '定义变量
Check = CBool(C)      ' Check 值为 False
```

CInt 函数可以将货币、单精度或双精度的数据转换成整型的数据。

```
Dim MyDouble, MyInt
MyDouble = 2134.8756 ' MyDouble 是 Double
MyInt = CInt(MyDouble) ' MyInt 值为 2134
```

VBA 中还有许多这样的函数，例如将表达式转换为 Long 子类型的 CLng 函数；将表达式转换为 Single 子类型 CSng 函数；将表达式转换为 String 子类型 CStr 函数，等等。读者可以在需要使用这些函数时查阅帮助文档，了解这些函数的具体用法。

技巧 2：自定义数据类型

在 VBA 中，用户可以使用 Type 语句自定义数据的类型，具体格式如下：

```
Type 数据类型名
    数据类型元素名 AS 数据类型
    数据类型元素名 AS 数据类型
End Type
```

其中，数据类型名为自定义类型的名称，数据类型为基本的数据类型。例如：

```
Type MyTpes
   Myname AS String '定义字符串存储一个名字
   Mysex AS Integer '定义整型存储性别
End Type
```

第 5 章　程序控制结构

利用 VBA 中的程序控制结构可以执行条件判断操作，也可以按条件的不同循环执行一些指定的操作，甚至还可以将条件判断和循环操作嵌套在一起使用。用户通过本章节的学习，将熟练掌握 VBA 编程中各种程序控制结构的知识。



5.1 VBA 程序结构概述

计算机只是按照人们的要求进行处理，所以开发人员需要告诉计算机在某些特殊条件下应该执行什么样的操作，VBA 中的程序结构的控制语句就是用来实现这样的控制功能的。常见的 VBA 中的程序结构控制语句分为如下几类：

（1）条件判断语句，包括 If Then Else 语句和 Select Case 语句。

（2）循环语句，包括 For Next 语句、Do loop 语句、While Wend 语句、For Each Next 语句。

（3）退出控制语句，包括 Go To 语句和 Exit 语句。



5.2 判断语句

条件表达式的结果取值只能是真（TRUE）或者假（FALSE），判断语句就是用来根据条件表达式的结果来控制程序的流程。在 VBA 中进行条件判断，最简单的是使用“If Then Else”语句，如果判断之后的结果很多，可以使用“Select Case”语句。

5.2.1 If Then Else 语句

VBA 中最常用的分支语句就是 If Then Else，该语句的用法很灵活，使用时根据业务需求可以分以下三种情况。

第一种是最简单的形式，只有一个条件判断分支语句，要么执行，要么不执行。语法格式如下：

```
If <条件表达式> Then <满足条件时需要执行的语句> End If
```

【例 5.1】使用 If Then 条件语句判断数值的大小，操作步骤如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-1 所示。

```
Sub Sample1()
  Dim num1 As Integer, num2 As Integer
  num1 = 5
  num2 = 10
  If num1 < 8 Then
      Debug.Print "num1 小于 8"
  End If
  If num2 < 8 Then
      Debug.Print "num2 小于 8"
  End If
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 5-2 所示。

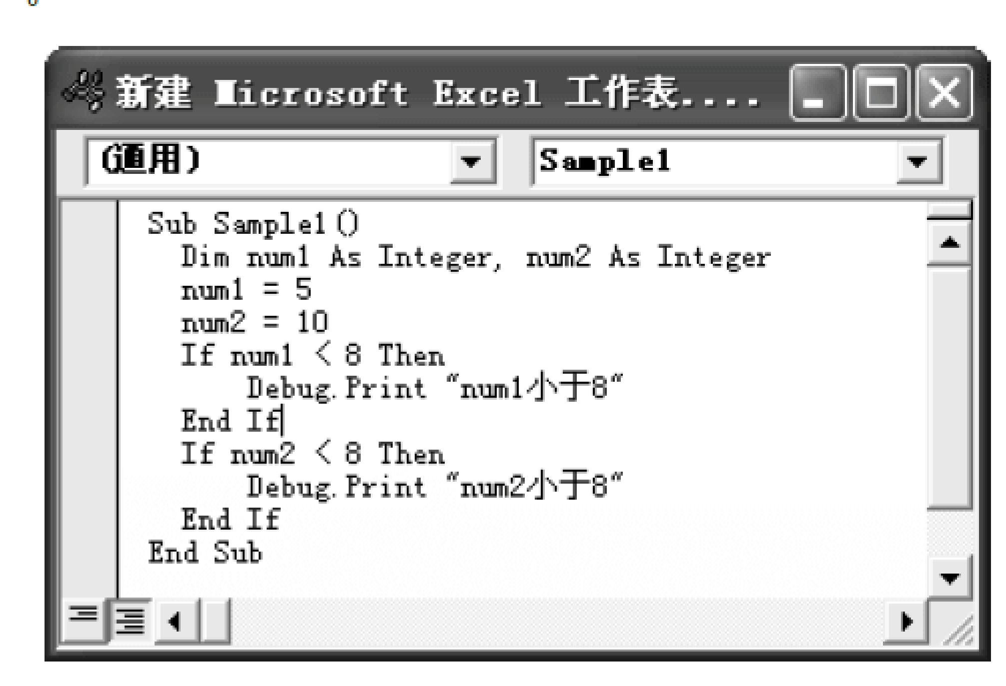


图 5-1 If Then 条件语句

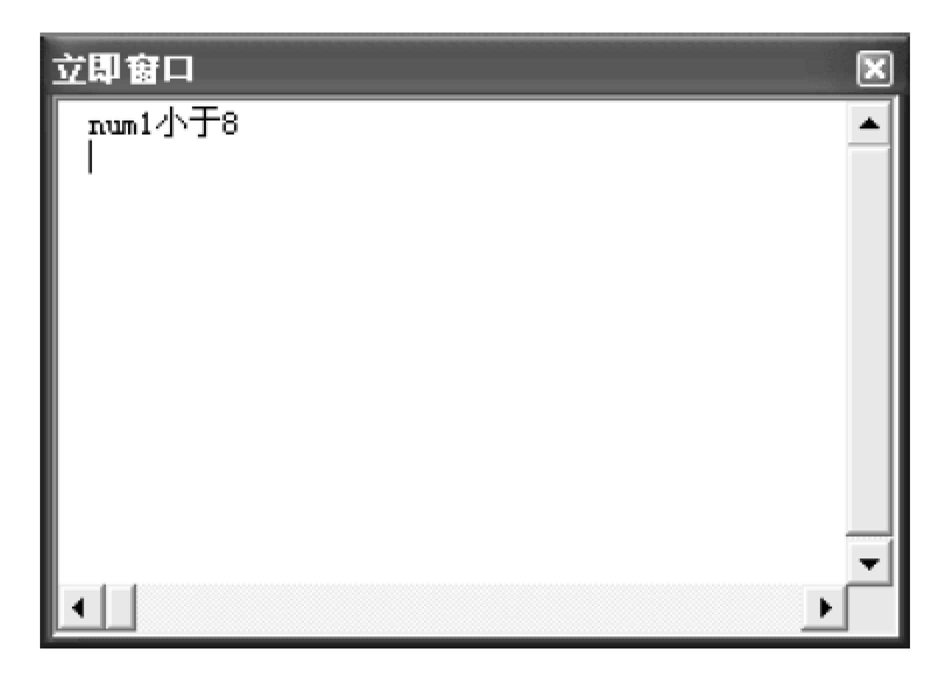


图 5-2 判断数值大小

Sample1 模块中定义了两个整型变量 num1 和 num2，num1 值为 5，num2 值为 10，第一个 If 语句判断 num1 的值是否小于 8，判断结果为真，所以打印输出结果字符串“num1 小于 8”；而 num2 的值是大于 8 的，所以条件判断结果为假，此时 Then 后面的语句不能被执行。

第二种形式是带有 Else 的形式：

```
If <条件表达式>  Then
<满足条件时需要执行的语句 1>
Else
<不满足条件时需要执行的语句 2>
End If
```

【例 5.2】使用 If Then Else 条件语句判断学生的成绩是否及格，操作步骤如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-3 所示。

```
Sub Sample2()
    Dim stuScore As Integer
    stuScore = 70
    If stuScore < 60 Then
        Debug.Print "成绩不及格"
    Else
        Debug.Print "成绩及格"
    End If
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 5-4 所示。

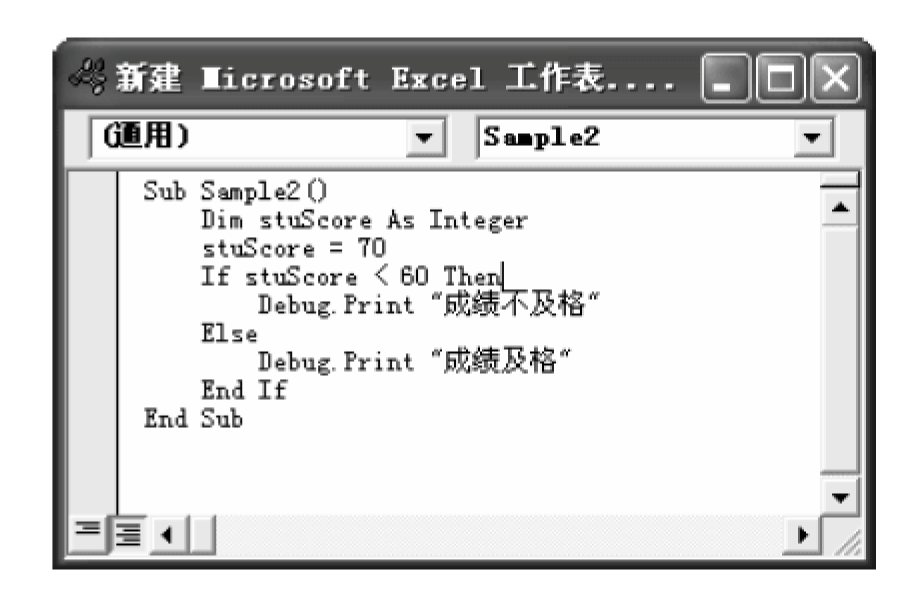


图 5-3　If Then Else 条件语句



图 5-4　If Then Else 语句执行结果

由结果可以看到，在 Sample2 模块中声明了一个变量 stuScore，该变量保存了一学生的成绩值，因为 stuScore 的值大于 70 分，所以 If 语句后面的条件表达式判断结果为 FALSE，所以程序流程将执行 Else 后面的语句。

第三种形式是多重条件的形式：

```
If <条件表达式 1> Then
<满足条件 1 时需要执行的语句 1>
ElseIf <条件表达式 2> Then
<满足条件 2 时需要执行的语句 2>
…
Else
<前面的条件都不满足时执行的语句 n>
End If
```

【例 5.3】使用 If 多重条件判断语句，根据学生的成绩发放奖学金，操作步骤如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-5 所示。

```
Sub Sampl3()
    Dim stuScore As Integer
    stuScore = 90
    If stuScore > 95 Then
        Debug.Print "该学生获得一等奖学金"
    ElseIf stuScore > 85 Then
        Debug.Print "该学生获得二等奖学金"
```

```
        Else
            Debug.Print "该学生未获得奖学金"
        End If
    End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 5-6 所示。




图 5-5　If 多重条件语句

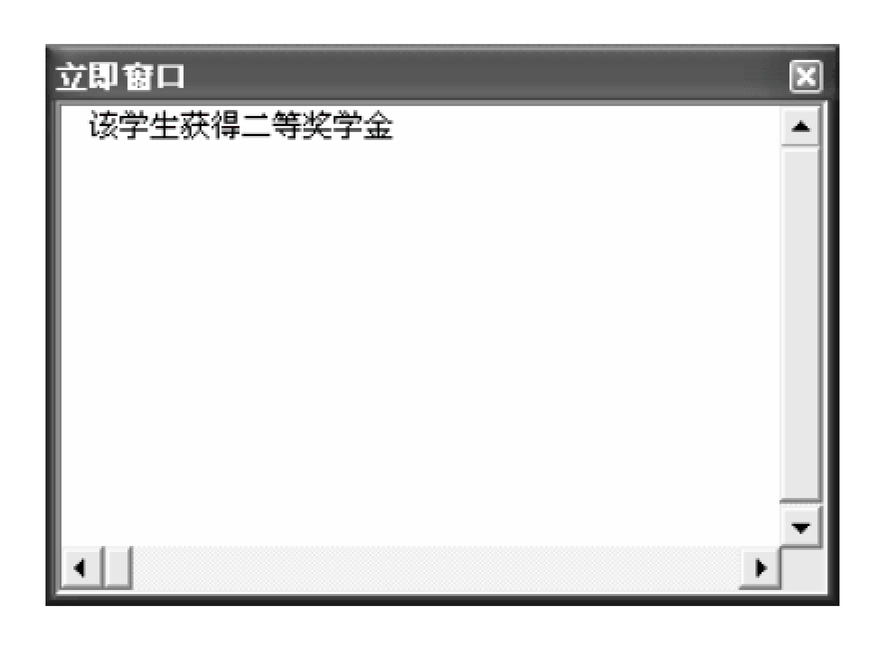


图 5-6　If 多重条件语句执行结果

由图 5-6 所示的执行结果可以看到，在 Sample3 模块中声明了一个变量 stuScore，该变量保存了一学生的成绩值，stuScore 的值大于 85 分但小于 95 分，所以 If 语句后面的条件表达式判断结果为 FALSE，接着程序流程将继续判断 ElseIf 后面的语句，表达式 stuScore > 85 的结果为 TRUE，所以打印输出后面的奖学金信息。

【提示】If 语句可以有多个分支，这些分支将依次进行判断，当某条件满足时，执行相应的语句，其余的分支将不再执行；若所有的条件都不满足，并且有 Else 子句，则执行 Else 语句后面的代码段，否则什么都不执行。

5.2.2 Select Case 语句

Select Case 语句也是用来进行多条件判断的。使用 Select Case 语句可以实现多重 If 语句相同的效果，但是使用 Select Case 语句编写的流程控制代码过程清晰，容易理解。在条件表达式的结果比较多时，可以选择使用多分支选择结果 Select Case 语句。

Select Case 语句根据条件表达式的求值结果，选择执行几个分支中的一个分支，其语法格式如下：

```
Select Case<条件表达式>
    Case  检验值 1
<代码段 1>
    Case  检验值 2
<代码段 2>
…
Case  检验值 n
<代码段 n>
Case Else
        <代码段 n+1>
```

```
End Select
```

这里的条件表达式与 If 中的条件表达式不同，If 条件表达式的结果只能是 TRUE 或者 FALSE，在 Select Case 语句中条件表达式的结果可以是数值或者是字符串。条件表达式的值与某个 Case 后面的值匹配，则执行相应的代码段；如果不止一个 Case 与条件表达式的值匹配，则只对第一个匹配的 Case 执行相应的代码段；如果所有的检验值没有一个与条件表达式的值匹配，则 VBA 执行 Case Else 语句后面的代码段。代码段中可以包含一条或多条代码。

【例 5.4】使用 Select Case 分支判断语句，操作步骤如下。

01 打开 Excel 工作簿，在表单 1 中 A 列下的表格中依次输入下面一些值：“一级作者”“二级作者”“三级作者”“四级作者”“五级作者”，操作数据位置如图 5-7 所示。

	A	B
1	一级作者	
2	二级作者	
3	三级作者	
4	四级作者	
5	五级作者	
6		

图 5-7　作者等级数据

02 打开 VBE，依次选择【插入】➤【模块】菜单命令。

03 在模块中输入以下代码，如图 5-8 所示。

```
Sub Sampl4()
  For Row = 1 To 5
    Select Case Range("A" & Row).Value
       Case "一级作者"
          Range("B" & Row).Value = 3000
       Case "二级作者"
          Range("B" & Row).Value = 3500
       Case "三级作者"
          Range("B" & Row).Value = 4000
       Case "四级作者"
          Range("B" & Row).Value = 4500
       Case "五级作者"
          Range("B" & Row).Value = 5000
    End Select
  Next
End Sub
```

04 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在 Excel 表单 1 中 B 列的前五行，如图 5-9 所示。

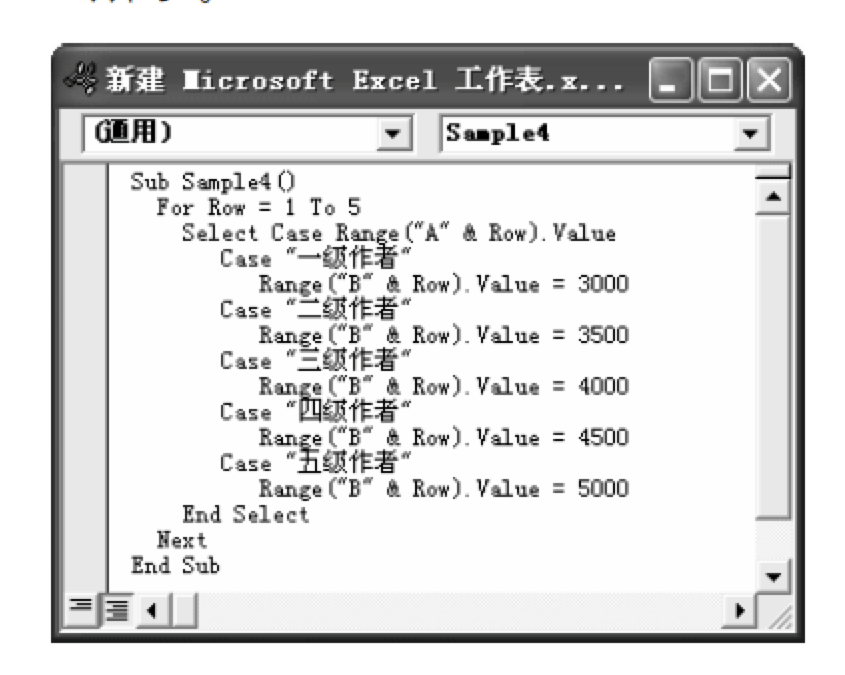


图 5-8　Select Case 分支语句

	A	B
1	一级作者	3000
2	二级作者	3500
3	三级作者	4000
4	四级作者	4500
5	五级作者	5000
6		

图 5-9　Select Case 语句执行结果

由图 5-9 执行结果可以看到，在 Sample4 模块中首先获取到表单 1 中 A 列各个表格中的值，Select Case 语句分别根据 A 列各个表格中的值，判断作者等级，然后将输出该等级的作者的稿费到 B 列中位于同一行的表格中。

上一个例子中，Select Case 中的 Case 检验值是一些具体的常量值，这些检验值还可以是一个表达式。

【例 5.5】根据购买产品的数量，决定出售的价格，操作步骤如下。

01 打开 Excel 工作簿，在表单 1 中 A 列下的表格中依次输入下面一些值：500,600,1000,300,5000,12000,100，操作数据位置如图 5-10 所示。

	A	B
1	500	
2	600	
3	1000	
4	300	
5	5000	
6	12000	
7	100	
8		

图 5-10　输入购买数量

02 打开 VBE，依次选择【插入】➤【模块】菜单命令。

03 在模块中输入以下代码，如图 5-11 所示。

```
Sub Sample5()
  Dim price AS Single,quantity AS Integer
  price = 100.0
  For Row = 1 To 7
    Quantity = Val(Range("A" & Row).Value)
    Select Case Quantity
       Case Is < 500
          Range("B" & Row).Value = price
       Case Is >= 500
          Range("B" & Row).Value = price * 0.95
       Case Is >= 500
          Range("B" & Row).Value = price * 0.9
       Case Is >= 1000
          Range("B" & Row).Value = price * 0.8
       Case Is >= 5000
          Range("B" & Row).Value = price * 0.8
       Case Is >= 10000
          Range("B" & Row).Value = price * 0.8
    End Select
  Next
End Sub
```

04 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在 Excel 表单 1 中 B 列的前五行，如图 5-12 所示。

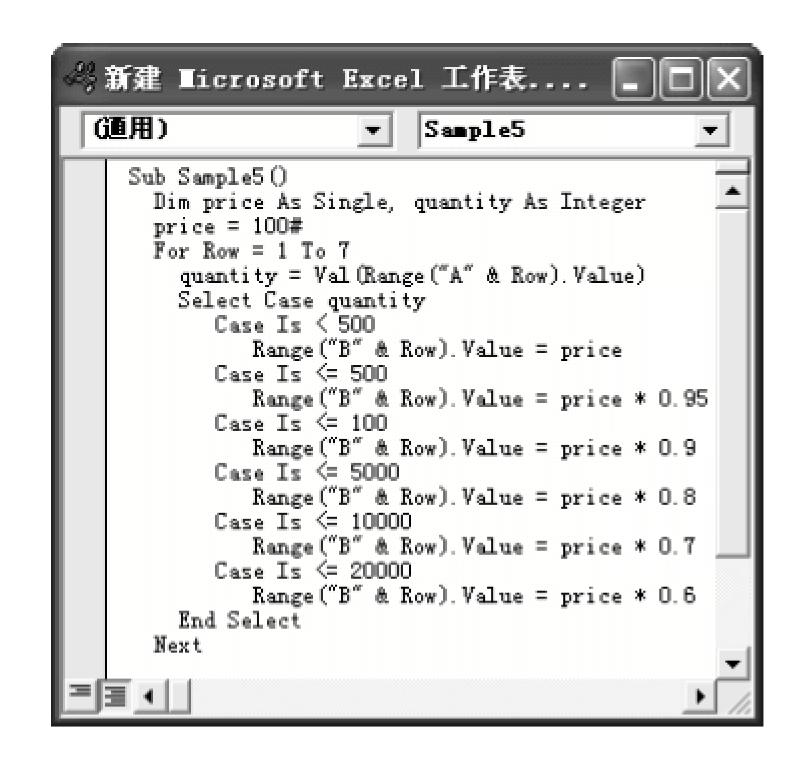


图 5-11　Sample5 过程代码

	A	B
1	500	95
2	600	80
3	1000	80
4	300	100
5	5000	80
6	12000	60
7	100	100
8		

图 5-12　Sample5 执行结果

由图 5-12 执行结果可以看到，在 Sample5 模块中首先获取到表单 1 中 A 列各个表格中的值，Select Case 语句分别根据 A 列各个表格中的值，判断购买的产品的数量，然后将输出该数量的打折后的价格到 B 列中位于同一行的表格中。



5.3 循环语句

前面章节中读者学习了什么是判断语句以及如何使用判断语句，在介绍 Select Case 语句的时候使用到了 For Next 语句，这是一个循环语句。在实际的业务中有时需要反复执行一段代码，这时就需要使用循环语句，VBA 中的循环语句有 For Next 语句、Do loop 语句、While Wend 语句和 For Each Next 语句。本节分别介绍这些语句的作用和用法。

5.3.1 For Next 语句

For Next 语句是最常见的一种循环语句，该语句将循环指定的次数，在执行循环操作时，For 循环中指定一个计数器，每次循环之后，计数器的数值会增加或者减少。For Next 语句的基本语法格式如下：

```
For 循环变量=初始值 To 终止值 [Step 步长]
    循环中执行的操作
Next [循环变量]
```

步长表示循环变量每次的增量，可以是正值也可以是负值，如果是正值，则初始值必须小于或等于终止值才能执行循环体；如果步长为负值，则初始值必须大于或等于终止值才能执行循环体，默认情况下步长值为 1。

【例 5.6】使用 For 循环将 A 列的偶数单元格背景设置交叉颜色，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-13 所示。

```
Sub Sample6()
    For Row = 1 To 50
        If Row Mod 2 = 0 Then
            Range("A" & Row).Interior.ColorIndex = 3
        Else
            Range("A" & Row).Interior.ColorIndex = 5
    End If
    Next Row
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在 Excel 表单 1 中 A 列的前 50 行，可以看到相邻两行的颜色交叉显示，如图 5-14 所示。

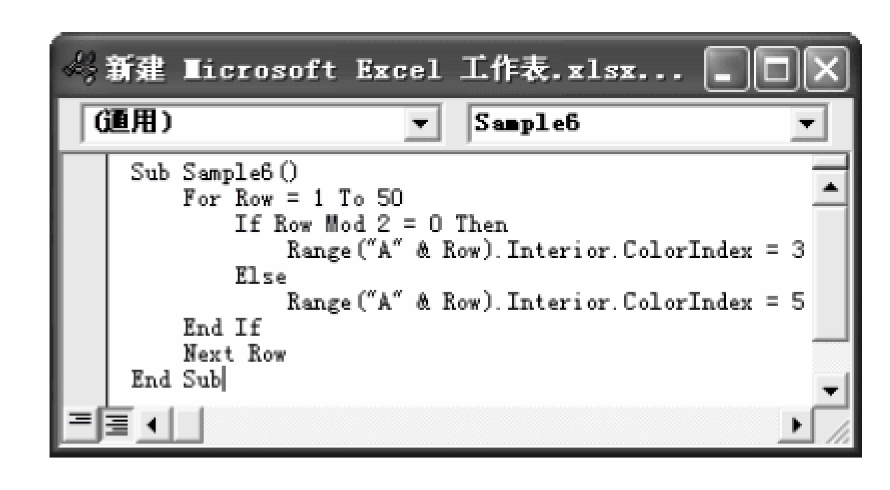


图 5-13　Sample6 过程代码

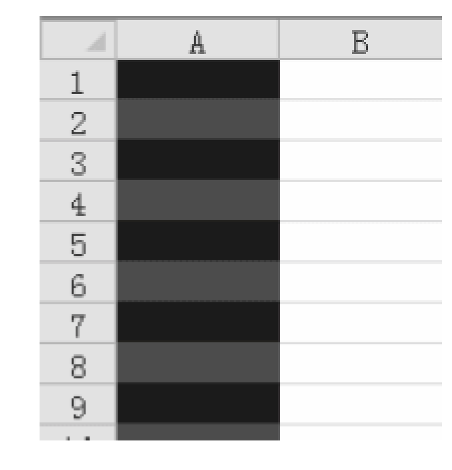


图 5-14　设置单元格颜色

在该循环中使用了一个 If 语句，该语句用于判断单元格所在格的行是偶数行还是奇数行，如果行号被 2 除时余数为 0，则表示这是偶数行，执行代码：

```
Range("A" & Row).Interior.ColorIndex = 3
```

否则是奇数行，执行下面代码：

```
Range("A" & Row).Interior.ColorIndex = 5
```

这里 Interior 表示单元格内部对象，而 ColorIndex 是内部对象中一个表示背景颜色的属性，通过该参数的值可以改变单元格的背景颜色。

【例 5.7】打印 For 循环中的循环变量值，指定步长为-2，操作过程如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-15 所示。

```
Sub Sample7()
    For Count = 10 To -2 Step -2
        Debug.Print Count
    Next Count
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出到【立即窗口】中，如图 5-16 所示。

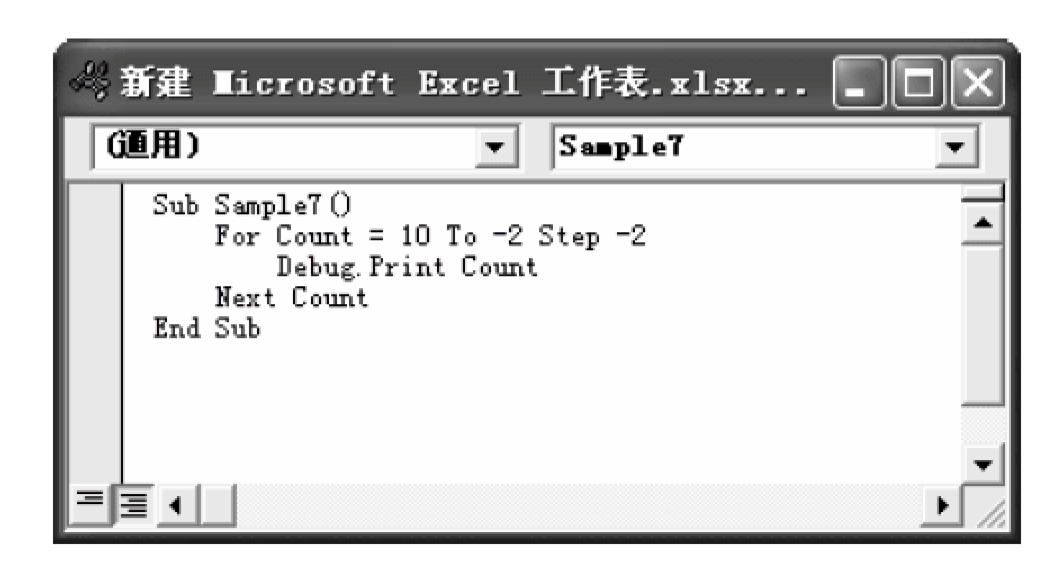


图 5-15　Sample7 过程代码



图 5-16　打印循环变量值

由图 5-16 的执行结果可以看到，循环变量的初始值为 10，终止值为-2，步长为-2，循环变量从 10 开始，每次减少 2，循环 6 次，每次可以得到一个不同的循环变量值，当循环变量值超过-2 时，循环终止。

5.3.2　Do Loop 语句

For Next 循环语句的循环次数是指定的，并且次数是确定的，而 Do Loop 循环语句不同，使用 Do Loop 语句循环执行一段代码时，循环的次数是不确定的。比如，公司的财务人员需要统计所有员工的工资，但是公司人员变动频繁，统计之前可能不知道员工的具体人数，此时就可以使用 Do Loop 循环。

Do Loop 语句需要指定循环条件，当条件为 True ，或直到条件变为 True 时，重复执行一个代码块。Do Loop 语句有两种用法，其语法格式分别是：

```
Do [{While | Until} condition]
[statements]
[Exit Do]
[statements]
Loop
```

或者可以使用下面这种语法：

```
Do
[statements]
[Exit Do]
[statements]
Loop [{While | Until} condition]
```

第一种语法是在进入循环之前检查条件表达式，第二种语法是在循环至少运行一次之后才检查条件表达式。

Condition 可以是一个数值表达式或字符串表达式，其值为 True 或 False，如果 Condition 是 Null，则 Condition 会被当做 False，Condition 是一个可选参数。

Statements 表示一条或多条命令，当或直到 Condition 为 True 时被重复执行。

Exit Do 语句用于跳出循环，Do...Loop 中可以在任何位置放置任意个数的 Exit Do 语句，随时跳出 Do...Loop 循环。Exit Do 通常用于条件判断之后，例如 If...Then，在这种情况下，

Exit Do 语句执行之后将转移到紧接在 Loop 命令之后的语句执行。

【例 5.8】提示 Loop 循环次数，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-17 所示。

```
Sub Sample8()
    counter = 0
    myNum = 20
    Do While myNum > 10
        myNum = myNum - 1
        counter = counter + 1
    Loop
    MsgBox "该循环执行了 " & counter & " 次! "
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将弹出一个对话框，告诉用户当前循环执行的次数，如图 5-18 所示。

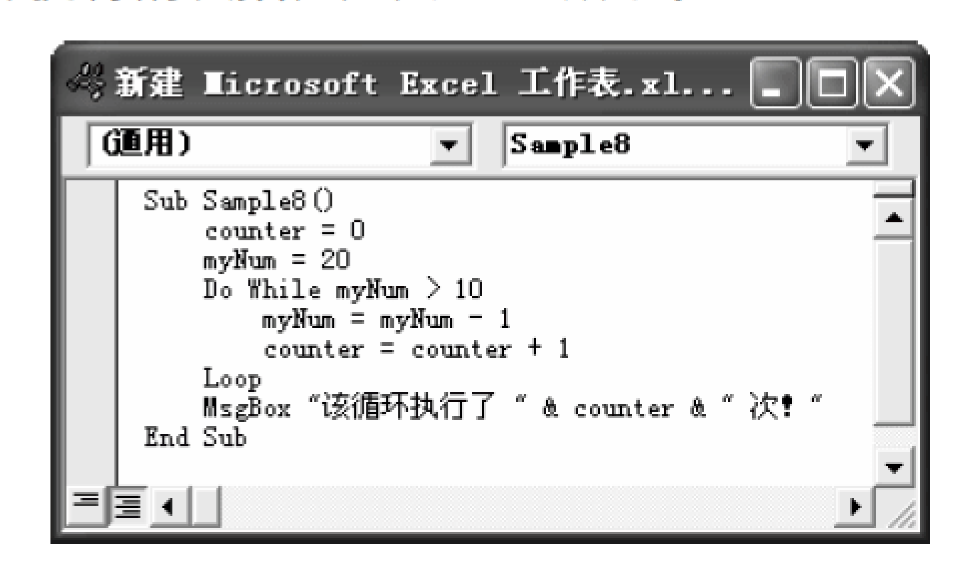


图 5-17　Sample8 过程代码



图 5-18　Loop 循环次数

该模块中定义了两个变量 counter 和 myNum，循环开始时首先对 myNum 进行判断，如果值大于 10，则执行 Do Loop 之间的代码块中的语句：myNum 值减 1，同时计数器 counter 加 1。循环结束时，将提示循环执行的次数。

下面介绍循环至少运行一次之后才检查条件表达式的 Do Loop 语句的用法。

【例 5.9】编写一个用户登录控制代码，如果用户输入的密码次数超过 3 次，禁止用户登录，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-19 所示。

```
Sub Sample9()
    Dim pwd As String
    Dim count As Integer
    Do
        pwd = InputBox("请输入密码")
        If pwd = "mypwd" Then
            Exit Do
        Else
```

```
            MsgBox ("请重新输入密码")
        End If
        count = count + 1
    Loop While count < 3
    If count >= 3 Then
        MsgBox ("超过登录次数，登录受限！")
    Else
        MsgBox ("欢迎登录")
    End If
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将弹出【请输入密码】对话框，告诉用户输入登录密码，如图 5-20 所示；如果输入的密码不是"mypwd"，则提示"请重新输入密码"消息，登录次数超过 3 次，用户将无法登录，如图 5-21 所示；如果密码正确，则提示"欢迎登录"的消息，如图 5-22 所示。

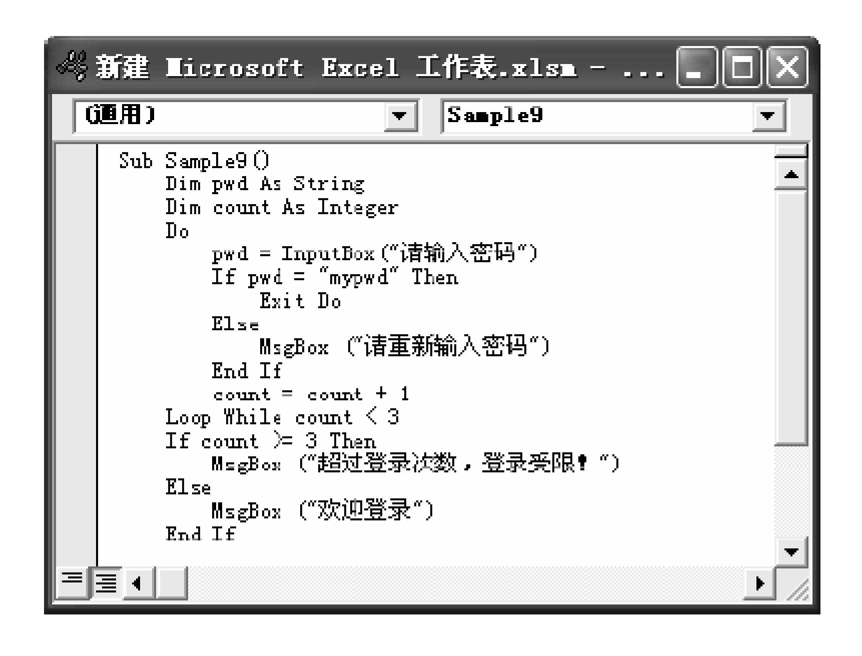


图 5-19　Sample9 过程代码



图 5-20　提示输入密码



图 5-21　登录失败



图 5-22　登录成功

5.3.3　While Wend 语句

While…Wend 语句是 Basic 早期版本中保留下的语句，其功能与 Do…While 循环语句相同，VBA 出于对早期版本的兼容性考虑，保留了该语句，其语法格式如下：

```
While <条件判断表达式>
<代码段>
Wend
```

如果 <条件判断表达式>为 True，则所有的代码段都会执行，一直执行到 Wend 语句。然后再回到 While 语句，并再一次检查 <条件判断表达式>，如果还是为 True，则重复执行。如果不为 True，则程序会从 Wend 语句之后的语句继续执行。

【例 5.10】While 循环变量用法，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-23 所示。

```
Sub Sample10()
    Dim Counter
    Counter = 0                          ' 设置变量初值。
    While Counter < 20                   ' 测试计数器的值。
        Counter = Counter + 1            ' 将计数器的值加一。
    Wend                                 '当 Counter > 19 时则循环终止。
    Debug.Print Counter                  ' 在“立即”窗口中显示数字
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将显示在【立即窗口】中，如图 5-24 所示。

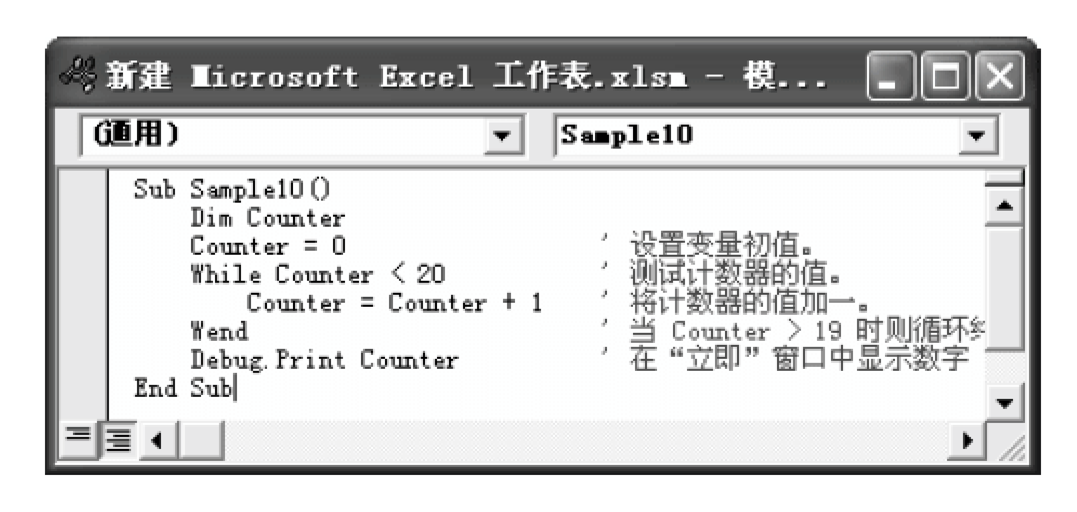


图 5-23　Sample10()过程代码



图 5-24　While 循环变量值

5.3.4　For Each Next 语句

For Each Next 语句与 For Next 语句看起来很像，For Next 语句执行循环前指定了循环的次数，而 For Each Next 语句在执行不确定循环次数。该语句是针对一个数组或集合中的每个元素，重复执行一组语句。For Each Next 语句的语法格式如下：

```
For Each <元素> In  <集合>
[代码段]
[Exit For]
Next [元素]
```

元素是用来遍历集合或数组中所有元素的变量。对于集合来说，元素可能是一个 Variant 变量、一个通用对象变量或任何特殊对象变量。对于数组而言，元素只能是一个 Variant 变量。

集合名称是对象所在的集合或数组的名称。如果集合中至少有一个元素，就会进入 For...Each 块执行。一旦进入循环，便先针对<集合>中第一个元素执行循环中的所有语句。如果<集合>中还有其他元素，则会针对它们执行循环中的语句，当<集合>中的所有元素都执行完了，便会退出循环，然后从 Next 语句之后的语句继续执行。

【例 5.11】遍历数组对象，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-25 所示。

```
Sub Sample11()
    Dim MyArray(10) As Integer, i As Variant
    For idx = 0 To 10
        MyArray(idx) = idx
    Next idx
    For Each i In MyArray
        Debug.Print i
    Next i
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将显示在【立即窗口】中，如图 5-26 所示。

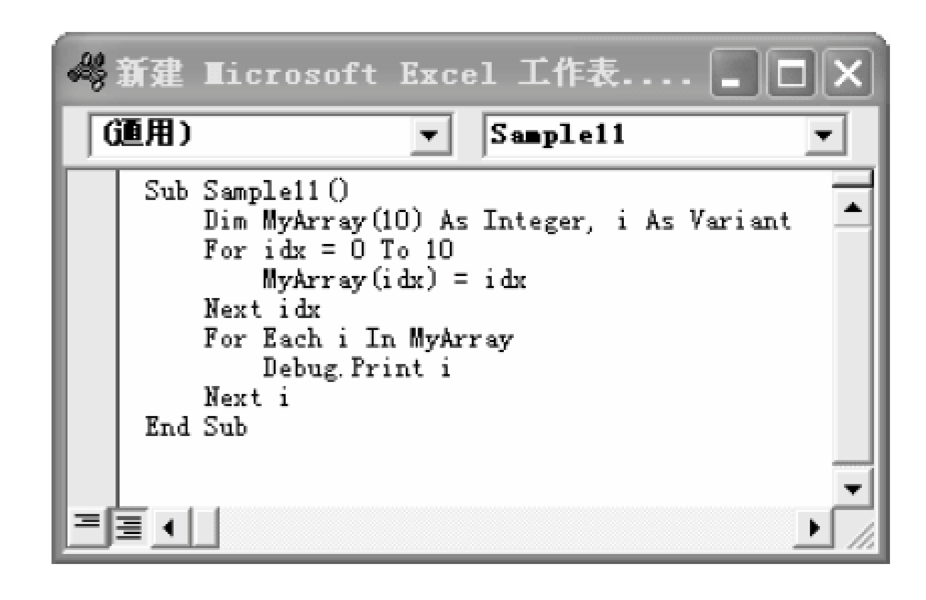






图 5-25　Sample11 过程代码　　　　图 5-26　遍历数组对象

该段代码首先使用 For Next 语句向 MyArray 数组中插入一些数值，然后使用 For Each Next 语句遍历数组中的各个对象的值。

【例 5.12】在指定 A1 到 C15 区域依次在每个表格中插入一个数值，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 5-27 所示。

```
Sub Sample12()
    Worksheets("Sheet1").Activate
    Dim i As Integer
    i = 1
    For Each obj In Range("A1:C15")
        obj.Value = i
        i = i + 1
    Next obj
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将显示 Excel 表单 1 中从 A1 到 C15 区域内，如图 5-28 所示。

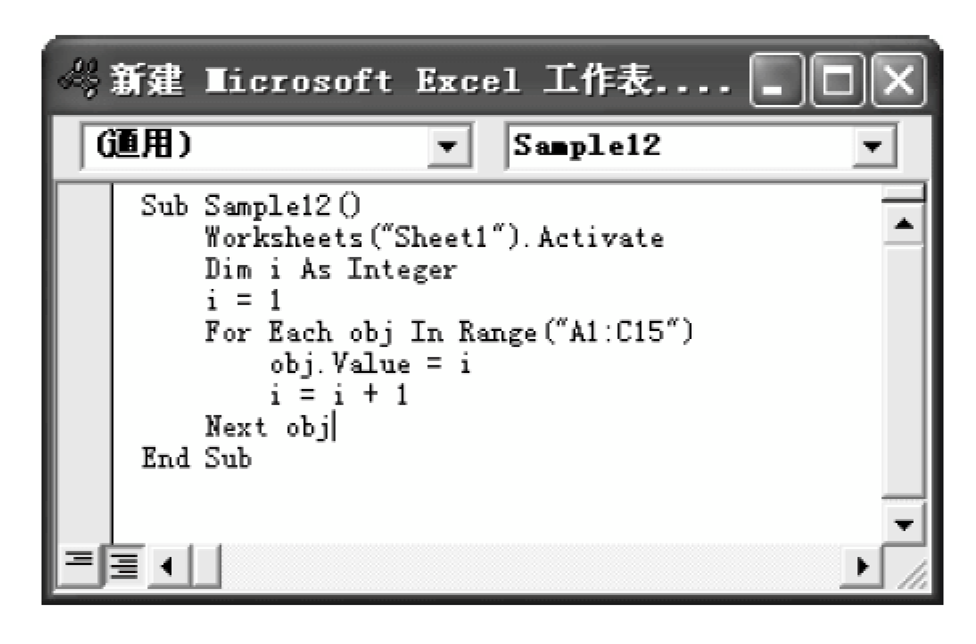



图 5-27　Sample12 过程代码

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4	10	11	12
5	13	14	15
6	16	17	18
7	19	20	21
8	22	23	24
9	25	26	27
10	28	29	30
11	31	32	33
12	34	35	36
13	37	38	39
14	40	41	42
15	43	44	45

图 5-28　向表格插入数据



5.4 退出控制语句

退出控制语句用于退出流程控制过程或跳到指定的语句位置处。退出控制语句包括 GoTo 语句和 Exit 语句。

5.4.1 GoTo 语句

GoTo 语句表示无条件地转移到过程中指定的行。其语法格式如下：

```
GoTo <行号/标号>
```

<行号/标号>参数可以是任意的行标签或行号。

GoTo 只能跳到它所在过程中的行。

【提示】GoTo 语句只能跳出所在过程中的行。

【例 5.13】设置跳转标签，使用 GoTo 语句在一个过程内的不同程序段间作流程控制，不同程序段用不同的“程序标签”来区隔，操作过程如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-29 所示。

```
Sub Sample13()
    Dim Number, MyString
    Number = Int(Rnd * 10) Mod 2        ' 设置变量初始值，值为 1 或者 0
    ' 判断 Number 的值以决定要完成哪一个程序区段（以“程序标签”来表式）。
    If Number = 1 Then GoTo Line1 Else GoTo Line2
Line1:
    MyString = "Number equals 1"
    GoTo LastLine    ' 完成最后一行。
Line2:
    MyString = "Number equals 2"
```

```
LastLine:
    Debug.Print MyString    ' 将“Number equals 1”显示在“立即”窗口。
End Sub
```

03 多次单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将显示在【立即窗口】中，如图 5-30 所示。

新建 Microsoft Excel 工作表.xlsm - 模块2 (代码)

(通用)　Sample13

```
Sub Sample13()
    Dim Number, MyString
    Number = Int(Rnd * 10) Mod 2        ' 设置变量初始值。
    ' 判断 Number 的值以决定要完成那一个程序区段(以“程序标签”来表式)。
    If Number = 1 Then GoTo Line1 Else GoTo Line2
Line1:
    MyString = "Number equals 1"
    GoTo LastLine      ' 完成最后一行。
Line2:
    MyString = "Number equals 2"
LastLine:
    Debug.Print MyString    ' 将“"Number equals 1"”显示在“立即”窗口。
End Sub
```

图 5-29　Sample13 过程代码

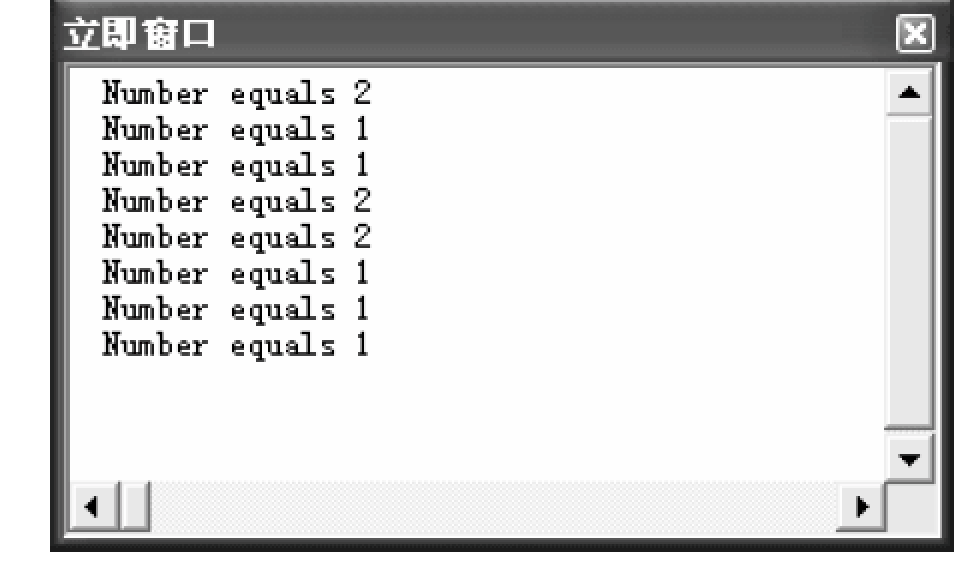


图 5-30　跳转流程控制

本段代码中根据 Number 的值来决定程序的跳转位置，当 Number 等于 1 时，执行“Line1:”标签后面的代码，将为 MyString 字符串变量赋值“Number equals 1”；当 Number 等于 2 时，将为 MyString 字符串变量赋值“Number equals 2”的消息，最后打印输出 MyString 字符串中的内容。可以看到两个赋值语句每次执行时只可能有一个语句被执行，这就是因为 GoTo 语句改变了程序的执行流程。使得程序可以根据条件跳转。

5.4.2 Exit 语句

Exit 语句用于当循环满足指定条件时强制退出循环，其语法格式如下：

```
Exit For|Do|Function| Sub
```

- Exit For 提供一种退出 For 循环的方法，并且只能在 For...Next 或 For Each...Next 循环中使用。Exit For 会将控制权转移到 Next 之后的语句。当 Exit For 用在嵌套的 For 循环中时，Exit For 将控制权转移到 Exit For 所在位置的外层循环。
- Exit Do 提供一种退出 Do...Loop 循环的方法，并且只能在 Do...Loop 循环中使用。Exit Do 会将控制权转移到 Loop 语句之后的语句。当 Exit Do 用在嵌套的 Do...Loop 循环中时，Exit Do 会将控制权转移到 Exit Do 所在位置的外层循环。
- Exit Function 立即从包含该语句的 Function 过程中退出。程序会从调用 Function 的语句之后的语句继续执行。
- Exit Sub 立即从包含该语句的 Sub 过程中退出。程序会从调用 Sub 过程的语句之后的语句继续执行。

【例 5.14】使用 Exit 语句退出 For...Next 循环、Do...Loop 循环及子过程，操作过程如下。

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 5-31 所示。

```
Sub Sample14()
    Dim I, MyNum
    Do                             ' 建立无穷循环。
        For I = 1 To 1000          ' 循环 1000 次。
            MyNum = Int(Rnd * 1000)    ' 生成一随机数码。
            Select Case MyNum          ' 检查随机数码。
                Case 7:
                    Debug.Print "Exit For"
                    Exit For       ' 如果是 7，退出 For...Next 循环。
                Case 29:
                    Debug.Print "Exit Do"
                    Exit Do        ' 如果是 29，退出 Do...Loop 循环。
                Case 54:
                    Debug.Print "Exit Sub"
                    Exit Sub       ' 如果是 54，退出子过程。
            End Select
        Next I
    Loop
    Debug.Print "退出过程！"
End Sub
```

03 多次单击工具栏中的【运行子过程/用户窗体】按钮，执行结果将显示在【立即窗口】中，如图 5-32 所示。

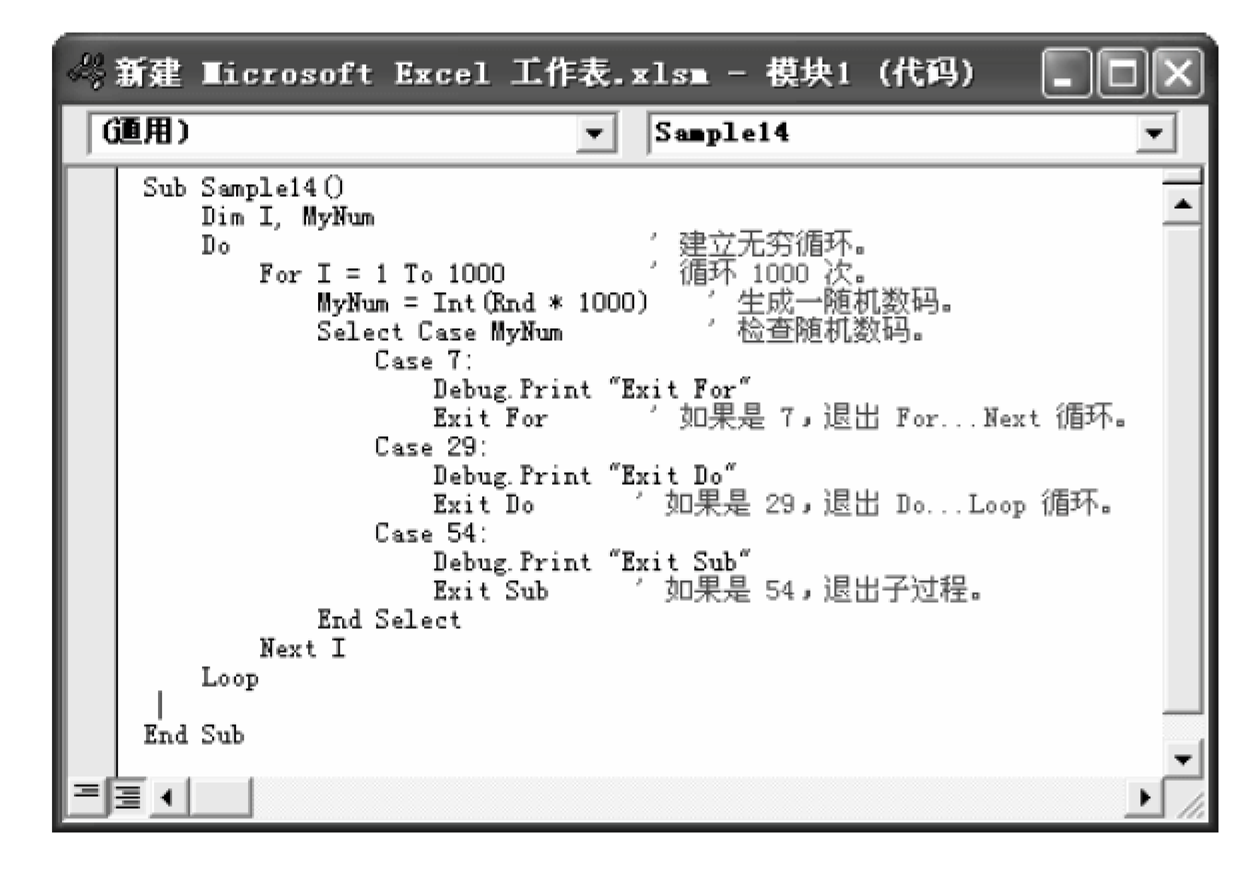


图 5-31　Sample14 过程代码

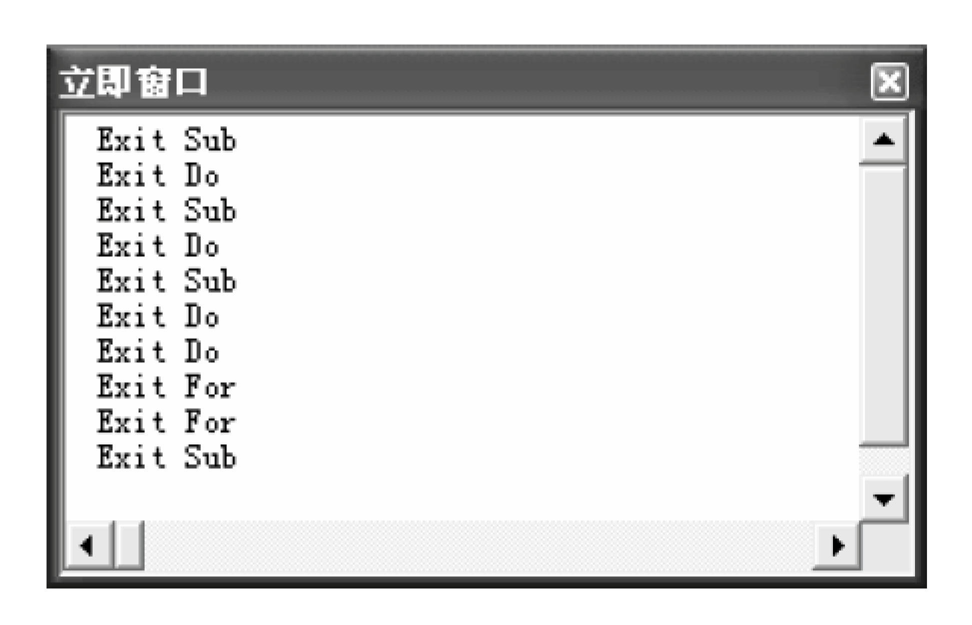


图 5-32　退出循环过程

由图 5-32 的结果可以看到，每次执行代码产生的结果都可能是不相同的，因为每次产生的随机数是不相同的，所以每执行一次代码，都有可能执行某一个 Exit 语句，退出某一层的循环，或者直接退出 Sub。



5.5 综合实战——评定学生成绩等级

学生成绩分为 5 个等级，分别根据每个学生的成绩进行登记判定，判断规则如下。

A：成绩大于或者等于 90 分。

B：成绩大于或者等于 80 分，小于 90 分。

C：成绩大于或者等于 70 分，小于 80 分。

D：成绩大于或者等于 60 分，小于 70 分。

E：小于 60 分。

操作步骤如下：

01 打开一个 Excel 表单，在名称为 Sheet1 的表单中输入下面一些数据，结果如图 5-33 所示。

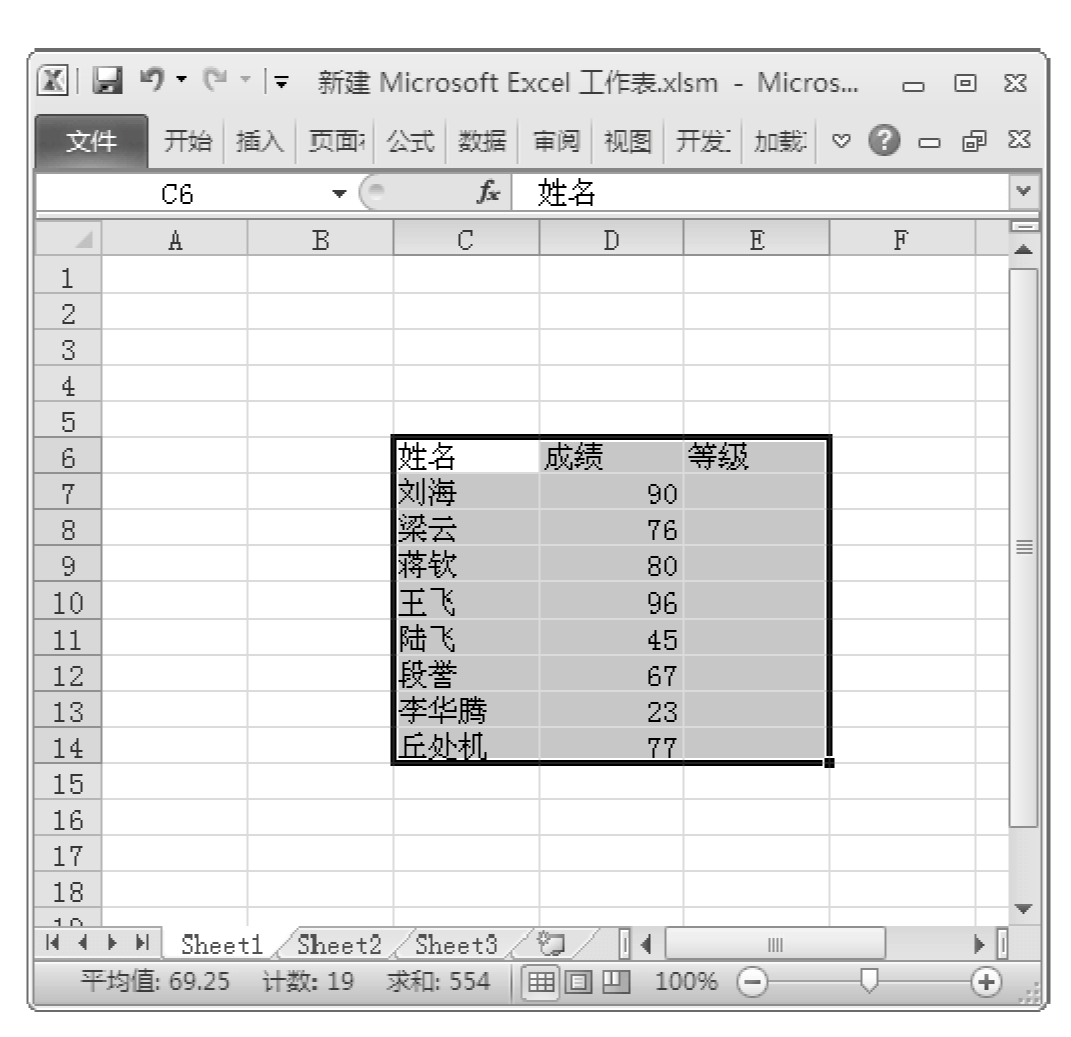


图 5-33　待评定学生成绩

02 按 Alt+F11 键，打开 VBE 环境，依次选择【插入】➤【模块】菜单命令，向工程中插入一个模块，编写代码如下：

```
Sub 判断等级()
    Dim iRow As Integer
    Dim sScore As Integer
    Dim level As String
    For iRow = 7 To 14
    Debug.Print Worksheets("Sheet1").Cells(iRow, 4).Value

        sScore = Worksheets("Sheet1").Cells(iRow, 4).Value
```




```
        If sScore >= 90 Then
            level = "A"
        ElseIf sScore >= 80 Then
            level = "B"
        ElseIf sScore >= 70 Then
          level = "C"
        ElseIf sScore >= 60 Then
            level = "D"
        Else: level = "E"
        End If
        Sheets(1).Cells(iRow, 5).Value = level
    Next iRow

End Sub
```

03 代码输入完成，单击【运行子过程/用户窗体】按钮，或者直接按 F5 键，执行模块 1 中的代码，执行之后可以看到不同学生的成绩等级信息显示在 E7 至 E14 的表格中，结果如图 5-34 所示。

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6			姓名	成绩	等级	
7			刘海	90	A	
8			梁云	76	C	
9			蒋钦	80	B	
10			王飞	96	A	
11			陆飞	45	E	
12			段誉	67	D	
13			李华腾	23	E	
14			丘处机	77	C	
15						

图 5-34　判断成绩等级



5.6 高手私房菜

技巧 1：合理利用 GoTo 语句

VBA 保留 GoTo 语句是为了保持与早期 Basic 版本的兼容，太多的 GoTo 语句，会使程序代码不容易阅读及调试，尽可能使用结构化控制语句。

技巧 2：Select Case 何时替换 If…Then…Else

Select Case 语句每次都要在开始时计算表达式的值；If…Then…Else 结构为每个 ElseIf 语句计算不同的表达式。只有当 If 语句和每一个 ElseIf 语句计算的表达式相同时，才能用 Select Case 结构替换 If…Then…Else 结构。

第 6 章　使用数组

数组是一种特殊的变量，该变量使用相同的名称存储一组数据。例如，需要所有员工的姓名和联系方式，用户可以创建两个变量来存储这两个内容，但使用数组可以同时存储这两项信息，而且可以一次保存上百个员工的信息。



6.1 数组简介

数组是一个集合，该集合中变量拥有相同的数据类型也可以拥有不同的数据类型，集合中的变量按照一定的顺序排列，使用数组名称和序号的组合可以指定访问数组中的某一个具体的数值。

根据其维数，数组可以分为一维数组、二维数组等更高维数的数组，常用的就是一维和二维数组，另外在 VBA 中还可以定义动态数组。与声明变量相似，声明数组时可以使用 Dim、Static、Private 或者 Public 语句来声明。



6.2 声明数组

声明数组时，要在数组名称后面加上括号，括号内指定数组包含的具体的元素数，紧接着括号后面可以写上元素的数据类型。下面将通过代码介绍声明数组的过程。

6.2.1 声明一维数组

声明长度固定的一维数组可以有两种方法，第一种是指定数组的上界；第二种是同时指定数组的上界和下界。

第一种方法的语法格式如下：

```
Dim 数组名称（上界） AS 数据类型
```

这里“上界”表示数组中序号的最大值，默认数组的下标序号从 0 开始到定义的上界。例如：

```
Dim FixArray(5) AS Double
```

这里数组名称为 FixArray，数组上界是 5，该数组中一共可以保存 6 个数据类型为 Double 的数据，分别是 FixArray（0）、FixArray（1）、...、FixArray（5）。

第二种方法的语法格式如下：

```
Dim 数组名 (下界 To 上界) AS 数据类型
```

该格式可以定义数组下标为一个任意的值，例如：

```
Dim strArray(-1 To 5) AS String
```

这里定义了一个数组 strArray，该数组中一共可以保存 6 个数据类型为 String 的数据，分别是 FixArray（-1）、FixArray（0）、...、FixArray（5）。

【提示】定义数组时，数组的下标必须是一个常数，不能是变量或者表达式。下面的代码是错误的，执行时将会报错，如图 6-1 所示。

```
Sub testArray()
 Dim lens As Integer
 lens = 5
 Dim MyArray(lens)
End Sub
```



图 6-1　错误的数组定义方式

在常用的编程语言中，例如 C、Java，数组初始下标都是从 0 开始的。在 VBA 中数组下标是否从 0 或 1 开始是根据 Option Base 语句的设置。如果希望数组索引从 0 开始，可以通过 Option Base 语句来设定，例如：

```
Option Base 1
Dim newArray(3)
```

这样 newArray 数组中将只包含 3 个元素，分别是 newArray(1)、newArray(2)和 newArray(3)。

【提示】Option Base 语句只能出现在用户窗体或模块的声明部分，不能出现在过程中，并且必须放在数组声明之前。

6.2.2 声明多维数组

前面介绍的数组是一维的，使用一个下标值就可以访问数组中的元素。多维数组就是数组的数组，即数组中的元素仍然是数组，读者可以把数据想象成 Excel 中一个局部的包含行和列的表格。在声明多维数组时要设定行和列的数值。声明多维数组可以有两种方法，其语法格式分别如下：

```
Dim 数组名称（第 1 维上界， 第 2 维上界） As 数据类型
```

和

```
Dim 数组名称（第 1 维下界 To 第 1 维上界，第 2 维下界 To 第 2 维上界） As 数据类型
```

如果需要将表单中的数据保存到二维数组中，可以使用下面的代码快速完成填充工作：

```
Dim sheetArray As Variant
sheetArray = Worksheets("Sheet1").Range("A2:D6")
```

多维数组中可以保存的数据个数是两个维数的乘积，多维数组可以是一个超出二维的数组，定义多维数组的格式与二维数组类似，每一维都使用逗号隔开即可。例如：

```
Dim tmpArray(1 To 300) As String
```

该语句定义了一个字符串数组，其中可以保存 300 个字符串。

```
Dim tmpArray(1 To 300, 1 To 10) As String
```

该语句定义了一个二维的字符串数组，其中可以保存 3000 个字符串，即 300×10。

```
Dim tmpArray(1 To 300, 1 To 10,1 To 20) As String
```

该语句定义了一个三维的字符串数组，其中可以保存 60 000 个字符串，即 300×10×20。

【提示】多维数组将会占用很大的内存空间，并且维数太多的数组并没有太多的实际价值，因为应用中很少使用超出三维的数组。



6.3 初始化数组

数组声明之后，其中还没有数据，VBA 在创建数组之后自动地初始化数组中的每一个元素：数值型的数组，其中每一个元素都将是 0 值，而字符串类型的数组，其中每个元素都是一个空字符串。本节将介绍为数组赋值的几种方法。

6.3.1 使用循环语句初始化数组

使用循环语句初始化数组是一种快速的方法，使用循环语句进行操作，可以让用户逐个输入数组的初始值，也可以按表单中的表格内存储的值来对数组赋值。

【例 6.1】为每一个数组元素指定值，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 6-2 所示。

```
Sub sample1()
    Dim idx As Integer
    Dim strArray(1 To 5) As String
    For idx = 1 To 5
        strArray(idx) = InputBox("请输入第" & idx & "个字母")
    Next
    For idx = 1 To 5
        Debug.Print strArray(idx)
    Next
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，系统弹出输入对话框，每次需要在这里输入一个字符串，输入完成之后数组中的元素将显示在【立即窗口】中，如图 6-3 所示。

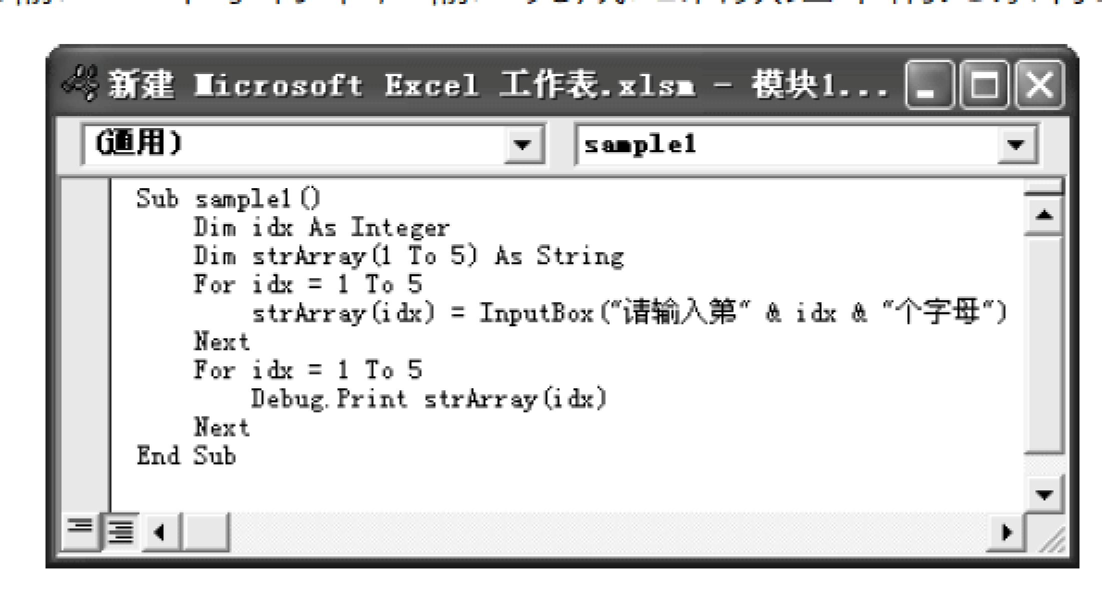


图 6-2　为数组元素指定值

图 6-3　打印数组元素

【例 6.2】使用数组中的值填充表单中的标题名称，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 6-4 所示。

```
Option Base 1

Sub Sample2()
    Dim titleArray As Variant
    Dim idx As Integer
    '声明用于填充表格标题名称数组
    titleArray = Array("员工姓名", "性别", "年龄", "编号")

    With Worksheets("Sheet1")        '选择 Sheet1 表单
```

```
        For idx = 1 To UBound(titleArray)    '遍历 titleArray 中的元素
            .Cells(1, idx).Value = titleArray(idx)  '设置表格标题名称
        Next count
    End With
End Sub
```

该段代码中创建数组时使用了 Variant 变量，该变量可以存储任何类型的信息，要快速填充数组，可创建一个类型为 Variant 的变量，这种变量可像数组那样使用，数据被存储到 Variant 变量中后，将具有数组的特征。

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果显示在 Excel Sheet1 中第一行的前 4 个表格，如图 6-5 所示。



图 6-4　Sample2 过程代码

	A	B	C	D	E
1	员工姓名	性别	年龄	编号	
2					
3					

图 6-5　填充表单中的标题名称

【例 6.3】将表单中的数组保存到二维数组，操作过程如下。

01 在上一个例题的基础上，读者首先在 Sheet1 表单中插入一些数据，例如笔者这里插入了一些员工的个人信息，如图 6-6 所示。

	A	B	C	D
1	员工姓名	性别	年龄	编号
2	李丽	女	21	101
3	李明	男	22	102
4	格林	男	23	103
5	露西	女	24	104
6	王磊	男	21	105
7				

图 6-6　员工个人信息数据

02 打开 VBE，依次选择【插入】➤【模块】菜单命令。

03 在模块中输入以下代码，如图 6-7 所示。

```
Sub Sample3()
    Dim sheetArray(2 To 6, 1 To 4)
```

```
    Dim row As Integer, col As Integer, count As Integer
    '将表单中的数据保存到二维数组中
    For row = 2 To 6
        For col = 1 To 4
            sheetArray(row, col) = Worksheets("Sheet1").Cells(row, col).Value
        Next col
    Next row
    '查看保存之后的结果
    For row = 2 To 6
        Debug.Print
        For col = 1 To 4
            Debug.Print sheetArray(row, col);
        Next col
    Next row
End Sub
```

04 单击工具栏中的【运行子过程/用户窗体】按钮，结果显示在【立即窗口】中，如图 6-8 所示。



图 6-7　将表单中的数据复制到二维数组

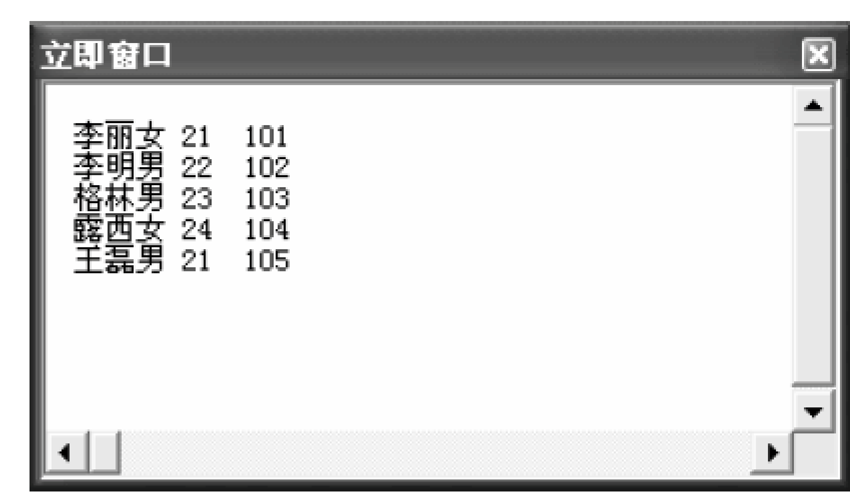


图 6-8　打印二维数组元素

由图 6-8 所示的结果可以看到使用二维数组可以完美地将表格中的数据保存起来。

6.3.2 使用 Array 函数初始化数组

使用刚才为数组赋值的方法，需要逐个输入元素值，因此效率很低。VBA 中可以使用 Array 函数为数组元素赋值，该函数将把数据集一次性插入到数组中，Array 函数的用法如下：

```
数组名称 = Array(数据集合)
```

数组名称是前面已经定好的数组名，函数中是一些常数构成的数据集合，多个值之间使用逗号隔开。

【例 6.4】使用 Array 函数为数组元素赋值，操作过程如下。

01 打开 VBE，依次选择【插入】➢ 【模块】菜单命令。

02 在模块中输入以下代码，如图 6-9 所示。

```
Sub Sample4()
    Dim monthArray As Variant, idx As Integer
    monthArray = Array("一月", "二月", "三月", "四月", "五月", "六月", "七月", "
八月", "九月", "十月", "十一月", "十二月")
      For idx = 0 To 11
           Debug.Print monthArray (idx)
      Next idx
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，如图 6-10 所示。



图 6-9　使用 Array 函数为数组赋值

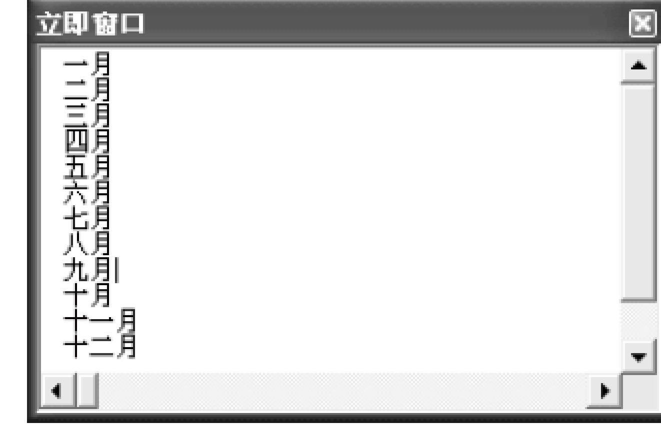


图 6-10　打印赋值结果

6.3.3 用数组值初始化数组

数组中的元素可以被访问和使用，所以，可以将一个数组中的元素复制到另一个数组中，使用一个循环语句就可以完成数组与数组的复制。

【例 6.5】使用数组元素对数组赋值，操作过程如下。

01 打开 VBE，依次选择【插入】➤ 【模块】菜单命令。

02 在模块中输入以下代码，如图 6-11 所示。

```
Option Base 1

Sub Sample5()
    Dim orignArray As Variant, destArray(1 To 12) As String
    Dim idx As Integer
    orignArray = Array("一月", "二月", "三月", "四月", "五月", "六月", "七月","
八月", "九月", "十月", "十一月", "十二月")
    '数组对拷
    For idx = 1 To UBound(orignArray)
        destArray(idx) = orignArray(idx)
    Next idx
    '打印目标数组中的元素
    For idx = 1 To UBound(destArray)
        Debug.Print destArray(idx)
    Next idx
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，与图 6-10 所示的结果相同。



图 6-11　使用数组元素为数组赋值

上面的代码中，如果 destArray 是一个 Variant 变量，那么通过一条语句就可以实现数组的对拷。Sample5 模块中的代码可以修改如下：

```
Option Base 1

Sub Sample5()
    Dim orignArray As Variant, destArray As Variant
    Dim idx As Integer
    orignArray = Array("一月", "二月", "三月", "四月", "五月", "六月", "七月", "
八月", "九月", "十月", "十一月", "十二月")
    '数组对拷
    destArray = orignArray
    '打印目标数组中的元素
    For idx = 1 To UBound(destArray)
        Debug.Print destArray(idx)
    Next idx
End Sub
```



6.4 动态数组

动态数组中可以保存的元素的个数是不确定的，前面介绍的数组在定义或初始化时已经限定了数组的长度，这些数组可以称为静态数组，而动态数组的长度在程序运行时可以改变。

6.4.1 声明动态数组

在使用动态数组之前，首先要声明。例如，下面的语句声明一个动态数组：

```
Dim dymArray()
```

该动态数组名称为 dymArray，其中将保持 Variant 类型的数据。

动态数组的使用分为两个步骤：第一步先在用户定义的模块或过程中使用 Dim 或者 Public 声明一个没有下标的数组，即括号内为空值；第二步在后面的代码中，每当需要时，可以使用 ReDim 语句重新定义该数组的大小。ReDim 语句的语法格式如下：

```
ReDim [Preserver] 数组名称(下边) [As 数据类型]
```

【例 6.6】获取当前 Excel 工作簿中各个表单的名称，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 6-12 所示。

```
Sub Sample6()
    Dim sheetArray() As String
    Dim idx As Integer, num As Integer
    num = ActiveWorkbook.Worksheets.count
    ReDim sheetArray(1 To num)
    For idx = 1 To num
        sheetArray(idx) = ActiveWorkbook.Sheets(idx).Name
    Next idx
    For idx = 1 To num
        Debug.Print sheetArray(idx)
    Next idx
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，如图 6-13 所示。

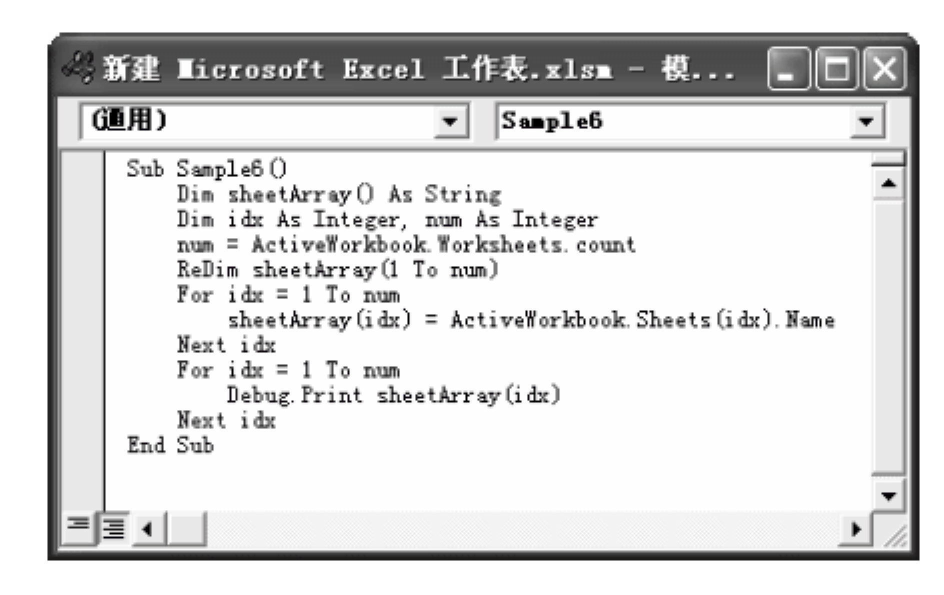




图 6-12　获取 Excel 表单的名称



图 6-13　打印各个表单名称

默认情况下，如果直接执行 ReDim 语句，则数组中原来的数据将会丢失，如果想要保持原先的值，则需要使用 ReDim Preserve 语句来扩充数组。如果是一个多维动态数组，则使用 Preserve 关键字只能改变最后一维的大小，并且不能改变维数的数目。例如，如果数组就是一维的，则可以重定义该维的大小，因为它是最末维，也是仅有的一维。不过，如果数组是二维或更多维时，则只有改变其最末维才能同时保留数组中的内容。

下面的示例介绍了如何在为已有的动态数组增加其最末维大小的同时而不清除其中所含的任何数据。

```
ReDim X(10, 10, 10)
. . .
ReDim Preserve X(10, 10, 15)
```

同样地，在使用 Preserve 时，只能通过改变上界来改变数组的大小，改变下界则会导致错误。如果将数组改小，则被删除的元素中的数据就会丢失。

6.4.2 数组的清除

动态数组可以在程序运行中改变数组的长度，那么静态数组是否也能改变呢？虽然 VBA 中不能改变静态数组的长度，但是可以在静态数组不需要时，将数组中的内容清除或对数组进行重定义，此时可以使用 Erase 语句来完成。Erase 语句的语法格式如下：

```
Erase 数组列表
```

数组列表是一个或多个用逗号隔开的需要清除的数组变量。

对于静态数组和动态数组，Erase 将采取完全不同的行为。

表 6-1 列出了 Erase 语句针对静态数组的操作规则。

表 6-1　Erase 对静态数组元素的操作

数组类型	Erase 对固定数组元素的影响
固定数值数组	将每个元素设为 0
固定字符串数组（长度可变）	将每个元素设为零长度字符串　("")
固定字符串数组（长度固定）	将每个元素设为 0
固定 Variant 数组	将每个元素设为 Empty
用户定义类型的数组	将每个元素作为单独的变量来设置
对象数组	将每个元素设为特定值 Nothing

而对于动态数组，Erase 释放动态数组所使用的内存。在下次引用该动态数组之前，程序必须使用 ReDim 语句来重新定义该数组变量的维数。

静态数组在使用 Erase 语句之后，仍然存在，只是其内容被清除了，而动态数组在使用 Erase 语句之后就不存在了。

【例 6.7】使用 Erase 语句清除数组，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 6-14 所示。

```
Sub Sample7()
    Dim NumArray(10) As Integer      ' Integer 数组。
    Dim StrVarArray(10) As String    '变长的 String 数组。
    Dim StrFixArray(10) As String     '定长的 String 数组。
    Dim VarArray(10) As Variant      'Variant 数组。
    Dim DynamicArray() As Integer    '动态数组。
```

```
    ReDim DynamicArray(10)     '分配存储空间。
    Erase NumArray        '将每个元素设为 0。
    Erase StrVarArray     '将每个元素设为
    '零长度字符串 ("")。
    Erase StrFixArray     '将每个元素设为 0。
    Erase VarArray     '将每个元素设为 Empty。
    Erase DynamicArray     '释放数组所用内存。
    'Debug.Print DynamicArray(1)     '动态数组被删除，访问将出错
    Debug.Print NumArray(1)        '静态数组被初始化，其元素值为 0
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出静态数组中元素的值，结果显示在【立即窗口】中，倒数第二个 Debug.Print 语句无法执行，执行将会出现“下标越界”的提示错误。如图 6-15 所示为错误提示信息框。



Microsoft Visual Basic

运行时错误 '9'：

下标越界

继续(C)　结束(E)　调试(D)　帮助(H)

图 6-14　Erase 语句清除数组　　　　图 6-15　访问错误



6.5 操作数组的函数

VBA 中提供了几个主要用于数组的函数：Array、IsArray、Erase、Lbound 和 Ubound，其中 Ubound、Array 和 Rease 函数在前面的例题中已经遇到过了。下面介绍另外两个函数的用法。

6.5.1 判断数组

数组和变量都可以使用 Dim 语句声明，有时仅根据变量的名称，无法判断这是一个普通变量还是一个数组变量，使用 IsArray 函数可以检查指定的变量是否为一个数组。如果是数组，则返回 True，否则返回 False。

【例 6.8】使用 IsArray 函数判断数组变量，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。



02 在模块中输入以下代码，如图 6-16 所示。

```
Sub Sample8()
    Dim nameArray(1 To 10) As Integer, ageArray, myCheck
    ageArray = Array(1, 2, 3)   '使用 Array 函数赋值

    myCheck = IsArray(nameArray)
    Debug.Print myCheck

    myCheck = IsArray(ageArray)
    Debug.Print myCheck

    myCheck = IsArray(myCheck)
    Debug.Print myCheck
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，如图 6-17 所示。

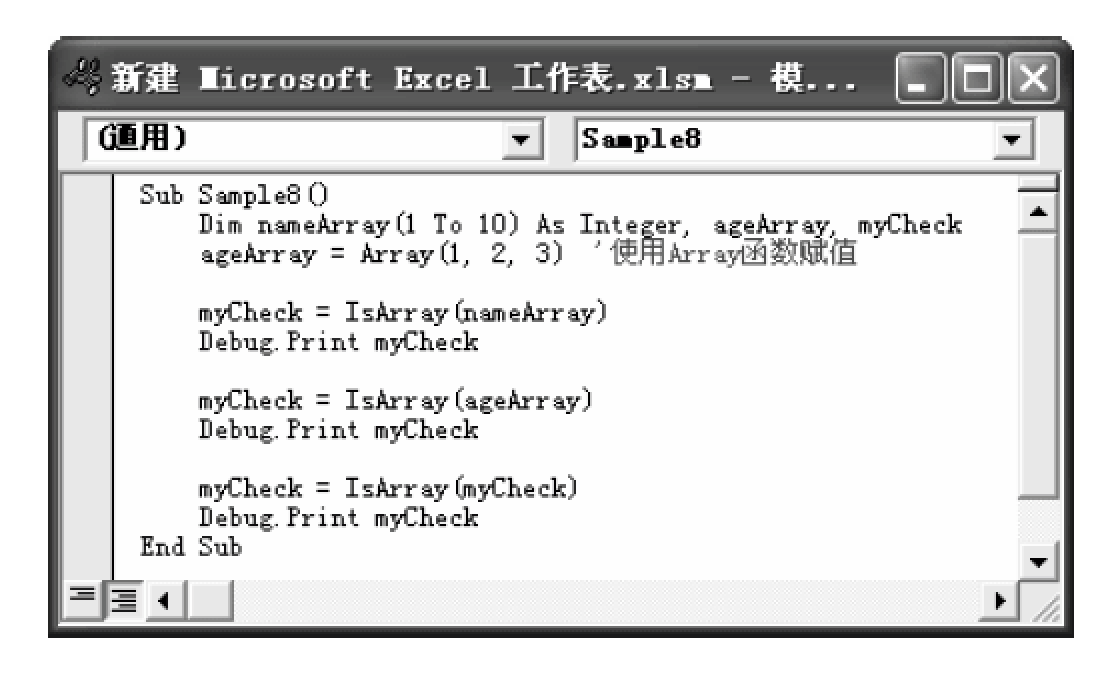


图 6-16　判断数组变量



图 6-17　IsArray 函数判断结果

6.5.2 查询数组的下标范围

LBound 函数和 UBound 函数对应，UBound 函数获取指定数组维的上界，而 LBound 函数获取指定数组维的下界。LBound 函数的语法格式如下：

```
LBound (数组名称[,维数])
```

数组名称是必需的。数组变量的名称遵循标准的变量命名约定。

维数是可选的，指定返回哪一维的下界。1 表示第一维，2 表示第二维，以此类推。如果省略维数，就认为是 1。

【例 6.9】使用 LBound 函数查询数组的下界值，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 6-18 所示。

```
Sub Sample9()
```

```
    Dim Lower
    Dim A(1 To 100, 0 To 3, -3 To 4)
    Dim AnyArray(10)

    Lower = LBound(A, 1) '返回 1
    Debug.Print Lower

    Lower = LBound(A, 2) '返回 0
    Debug.Print Lower

    Lower = LBound(A, 3) '返回-3
    Debug.Print Lower

    Lower = LBound(AnyArray)    '返回 0 或 1，取决于
                               'Option Base 的设置。
    Debug.Print Lower
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，如图 6-19 所示。



图 6-18　查询数组下界



图 6-19　打印下界值



6.6 综合实战——对数组中的随机数据进行排序

实际应用中经常会遇到需要对某列数据进行降序或者升序的方式排列，这两个过程的思路是相同的，这里以降序排列为例。

操作步骤如下：

01 打开一个 Excel 表单，按 Alt+F11 键，打开 VBE 环境，依次选择【插入】➤【模块】菜单命令，向工程中插入一个模块，编写代码如下：

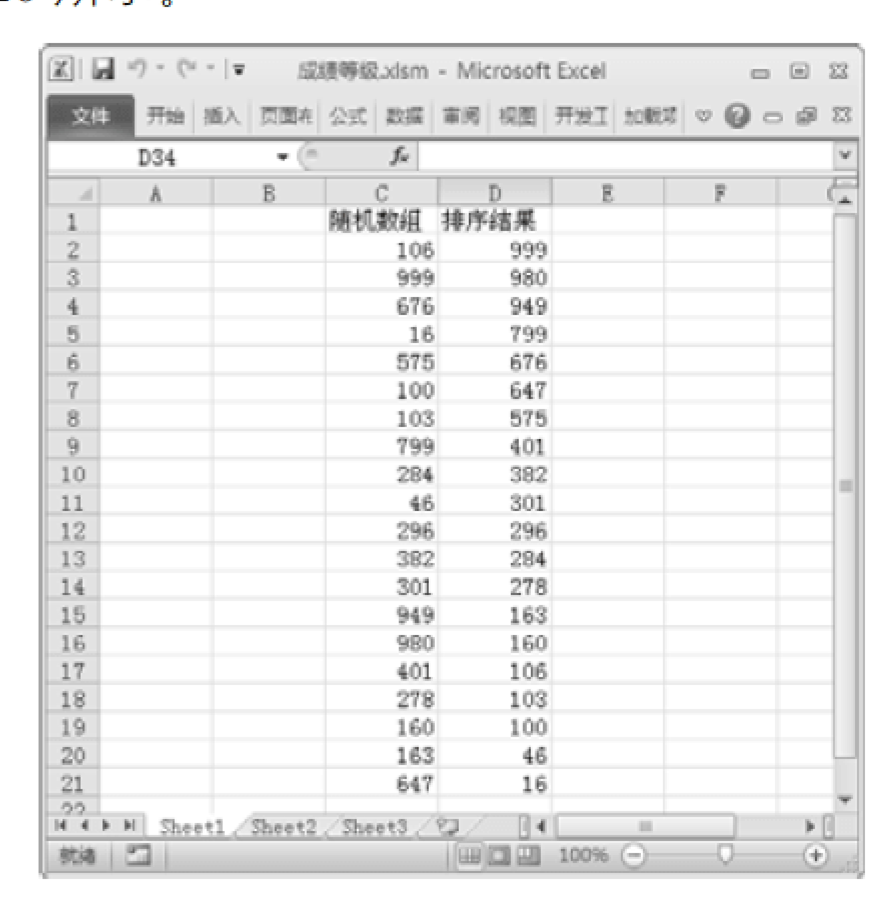
```
Sub getMaxValue()
```

```
    Dim myArray(1 To 20) As Integer
    Dim idx As Integer
    Worksheets("Sheet1").Cells(1, 3).Value = "随机数组"
    Worksheets("Sheet1").Cells(1, 4).Value = "排序结果"

    For idx = 1 To 20
        myArray(idx) = Rnd * 1000 '生成 0 到 1000 之间的一个随机数
        Worksheets("Sheet1").Cells(idx + 1, 3).Value = myArray(idx)
    Next
    '对数组中数据进行排序
    For idx = 1 To 19
        For j = idx + 1 To 20
            If myArray(idx) < myArray(j) Then
                tempVar = myArray(idx)
                myArray(idx) = myArray(j)
                myArray(j) = tempVar
            End If
        Next j
    Next idx
    '显示排序之后的数组
    For idx = 1 To 20
        Worksheets("Sheet1").Cells(idx + 1, 4).Value = myArray(idx)
    Next

End Sub
```

02 代码输入完成，单击【运行子过程/用户窗体】按钮，或者直接按 F5 键，执行模块 1 中的代码，执行之后可以看到 C2 至 C21 表格中是待排序的一个随机数组，D2 至 D21 表格中则存储了排序之后的结果，如图 6-20 所示。



	C	D
1	随机数组	排序结果
2	106	999
3	999	980
4	676	949
5	16	799
6	575	676
7	100	647
8	103	575
9	799	401
10	284	382
11	46	301
12	296	296
13	382	284
14	301	278
15	949	163
16	980	160
17	401	106
18	278	103
19	160	100
20	163	46
21	647	16

图 6-20 对随机数组进行排序



6.7 高手私房菜

本章主要介绍了 Excel 2010 VBA 中数组的相关内容，包括如何声明一维和多维数组，如何初始化数组，如何创建动态数组和操作数组函数。下面继续来解答一些常见的疑惑。

技巧 1：数组元素的序号规则

数组元素的序号是从 0 开始，数组下界是从 0 还是从 1 开始是根据 Option Base 语句设置的，如果没有指定 Option Base 为 1，则数组下界默认从 0 开始。

技巧 2：清除数组需注意

使用 Erase 语句可以轻松清除数组，重新初始化固定的数组的元素，以及释放动态数组的存储空间。清除数组时需注意，只能给出数组名，不能带括号和下标。

例如，Erase dymArray（）是错误的写法。

技巧 3：利用 Array 函数为数组赋值

Array 函数可以为数组赋值，但是 Array 函数返回值是 Variant 类型，所以在使用 Array 函数给数组变量赋初值之前，定义数组时只能将其定义为 Variant 类型，且不能设置下标，否则将显示“不能给数组赋值”的错误信息。

第 7 章　使用过程

过程是 VBA 中的基本程序功能单元，VBA 应用程序就是由许许多多个过程组成的。Excel VBA 中的过程分为通用过程和事件过程。通用过程又可以分为两种类型，一种是 Sub 过程；一种是 Function 过程，也就是自定义函数。本章将主要介绍如何在 VBA 定义和调用过程。



7.1 过程的相关概念

在模块内的代码会被组织成过程，而过程会告诉应用程序如何去执行一个特定的任务，利用过程可将复杂的代码细分成许多部分，以便管理。VBA 中的主要任务就是编写过程，除了系统提供的事件过程和内部函数过程外，用户可以自定义下面四种过程：

- Sub 子过程
- Function 函数过程
- Property 属性过程
- Event 事件过程

使用过程可以简化复杂程序的执行过程，提高编写代码的效率，便于调试和维护。



7.2 定义 Sub 过程

定义 Sub 过程有两种方式，一种是使用对话框定义子过程；另一种是在模块中直接输入代码。

7.2.1 使用对话框定义子过程

使用对话框定义子过程就是提供了一个定义子过程的向导模板，用户在定义时只用指定子过程的名称、类型和使用范围，就可以生产一个子过程的模板代码。

【例 7.1】使用对话框定义子过程，操作过程如下。

01 在 VBE 环境中首先选择【插入】➤【模块】菜单命令，新建一个模块。

02 再次选择系统菜单中的【插入】➤【过程】菜单命令，打开【添加过程】对话框，在其中可以设定过程的名称，如图 7-1 所示。



图 7-1　【添加过程】对话框——子程序

【添加过程】对话框中的主要参数含义如下：

- 【类型】选项框：可以选定该过程的类型，包括子程序、函数和属性。
- 【范围】选项框：可以选择过程的有效范围。【公共的】表示创建的过程在全局各个模块中都可以调用，在过程定义代码中将添加 Public 关键字；【私有的】表示过程只能在局部使用。
- 【把所有局部变量声明为静态变量】复选框：选中该选项，将在过程名称前面添加 Static 关键字。

01 设置完各个参数之后，单击【确定】按钮，VBA 将在模块窗口中自动生成过程的结构代码，如图 7-2 所示。

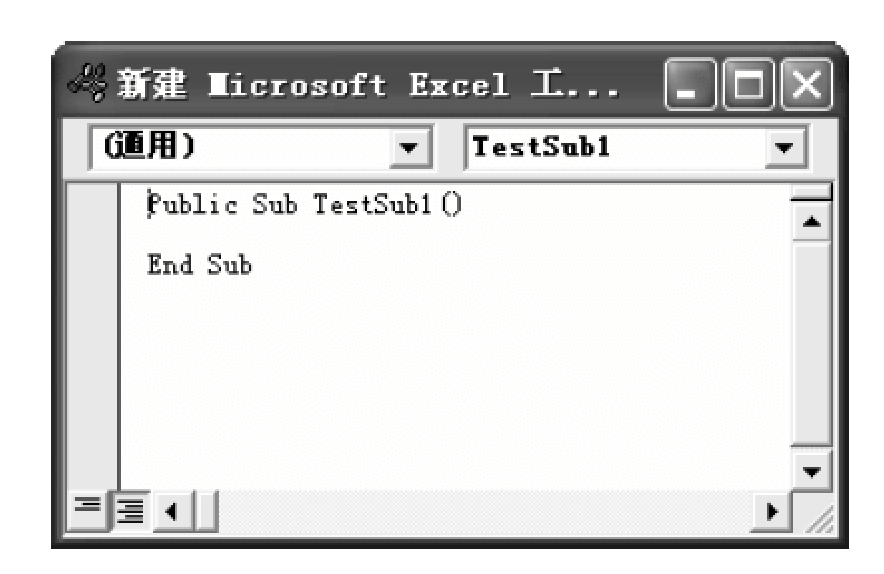


图 7-2　生成子过程结构代码

【提示】每个 Sub 过程都必须有一个 End Sub 语句，End Sub 表示当程序执行到这里就结束该过程的运行。不过，可以使用 Exit Sub 语句结束跳出过程的执行。Sub 过程的结构已经搭建好了，下面可以向过程中添加代码了。

01 在该过程结构中插入以下代码，打印输出九九乘法表。

```
Public Sub TestSub1()
Dim i As Integer
   Dim j As Integer
   '通过循环的嵌套实现九九乘法表
   For i = 1 To 9
     For j = 1 To i
        Debug.Print Tab((j - 1) * 9 + 1); i & "×" & j & "=" & i * j;
     Next j
   Next i
End Sub
```

02 单击工具栏中的【运行子过程/用户窗体】按钮，打印输出数组中元素的值，结果显示在【立即窗口】中，如图 7-3 所示。

```
立即窗口
1×1=1
2×1=2   2×2=4
3×1=3   3×2=6   3×3=9
4×1=4   4×2=8   4×3=12  4×4=16
5×1=5   5×2=10  5×3=15  5×4=20  5×5=25
6×1=6   6×2=12  6×3=18  6×4=24  6×5=30  6×6=36
7×1=7   7×2=14  7×3=21  7×4=28  7×5=35  7×6=42  7×7=49
8×1=8   8×2=16  8×3=24  8×4=32  8×5=40  8×6=48  8×7=56  8×8=64
9×1=9   9×2=18  9×3=27  9×4=36  9×5=45  9×6=54  9×7=63  9×8=72  9×9=81
```

图 7-3　九九乘法表

7.2.2 使用代码创建 Sub 过程

用户不仅可以使用对话框来创建 Sub 过程的框架，还可以直接手动输入代码来创建，对于初学者来说，这样可以了解到 VBA 到底做了些什么，并且可以加深对创建 Sub 过程的理解。创建 Sub 过程的语法结构如下：

```
[Private | Public | Friend] [Static] Sub <过程名称> [(参数列表)]
[语句段]
[Exit Sub]
[语句段]
End Sub
```

- Public 是可选参数，表示所有模块的所有其他过程都可访问这个 Sub 过程。 如果在包含 Option Private 的模块中使用，则这个过程在该工程外是不可使用的。
- Private 是可选参数，表示只有在包含其声明的模块中的其他过程可以访问该 Sub 过程。
- Friend 是可选参数，只能在类模块中使用。表示该 Sub 过程在整个工程中都是可见的，但对于对象实例的控制者是不可见的。
- Static 是可选参数，表示在调用期间保留 Sub 过程的局部变量的值。Static 属性对在 Sub 外声明的变量不会产生影响，即使过程中也使用了这些变量。
- <过程名称>指定 Sub 的名称。
- [（参数列表）]是可选的，是指在调用时要传递给 Sub 过程的参数的变量列表，多个变量则用逗号隔开。

● [语句段]中包含 Sub 过程中所执行的任何语句组。

下面看一下如何手动地在代码窗口中创建 Sub 过程。

【例 7.2】手动创建 Sub 过程，操作过程如下：

01 在 VBE 环境中首先选择【插入】➢【模块】菜单命令，新建一个模块或者直接双击【工程资源管理器】窗口中的某一个模块，打开代码窗口。

02 在代码窗口中输入 Sub Testsub2 语句，然后按回车键，系统将自动添加后面的括号和 End Sub 语句，结果如图 7-4 所示。

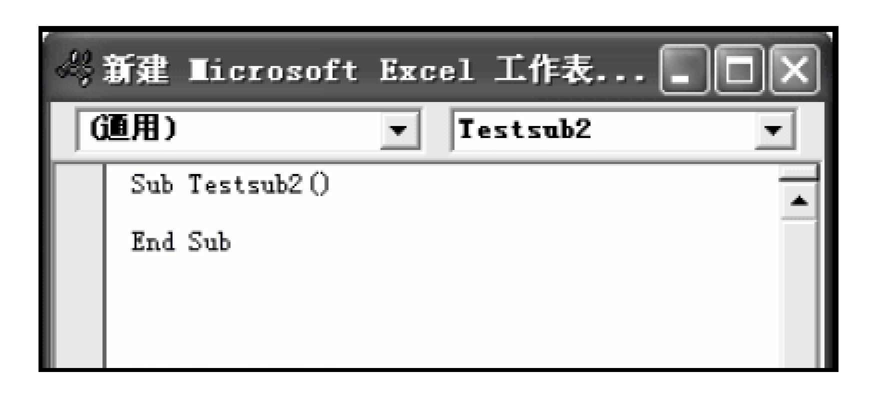
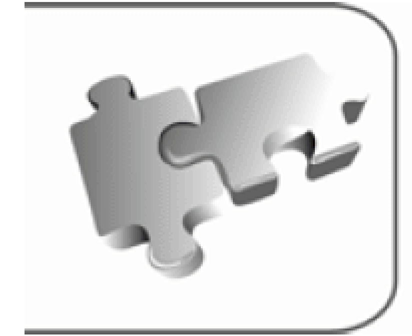


图 7-4　手动创建子过程结构代码

03 至此，就完成了手动创建 Sub 过程的操作。



7.3 定义 Function 函数过程

Function 是 VBA 中的另一种通用过程类型，可以将其称为函数。Function 过程的定义与 Sub 过程的定义有相同点和不同点。但是 Function 过程可以在程序调用时直接返回一个值。本节将介绍如何定义 Function 过程。

7.3.1 使用对话框定义函数过程

创建函数也有两种方法：使用对话框定义函数或者手动输入代码，这两种方法与创建 Sub 过程的操作步骤基本一样。

【例 7.3】使用对话框定义函数，操作过程如下：

01 在 VBE 环境中首先选择【插入】➢【模块】菜单命令，新建一个模块或者直接双击【工程资源管理器】窗口中的某一个模块，打开代码窗口。

02 再次选择系统菜单中的【插入】➢【过程】菜单命令，打开【添加过程】对话框，在其中选中【函数】单选按钮，如图 7-5 所示。

03 设置完各个参数之后，单击【确定】按钮，VBA 将在模块窗口中自动生成 Function 过程的结构代码，如图 7-6 所示。




图 7-5 【添加过程】对话框——函数

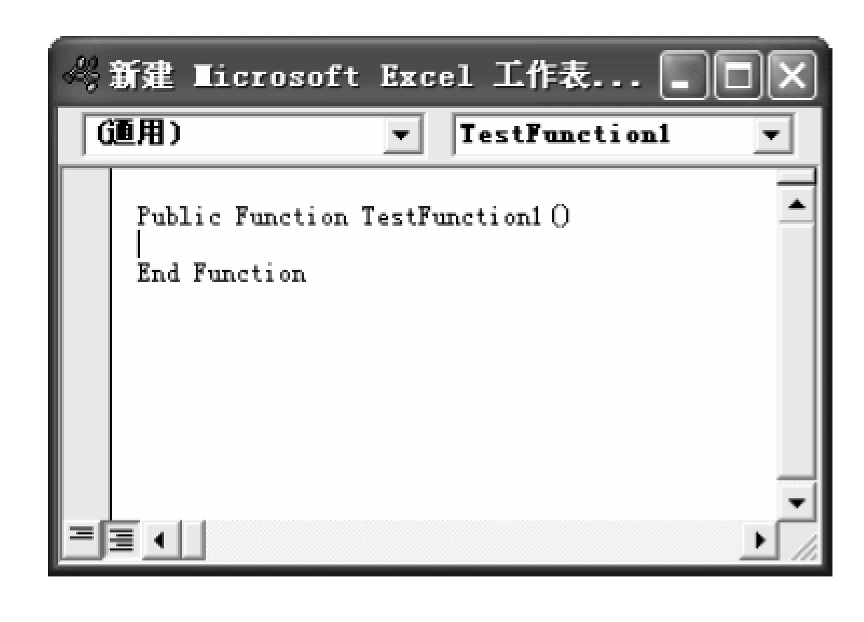


图 7-6 生成函数结构代码

7.3.2 使用代码创建 Function 过程

同样用户可以手动添加创建 Function 的代码，创建 Function 过程的基本语法结构如下：

```
[Public | Private | Friend] [Static] <函数名称> [(参数列表)] [As 数据类型]
[语句段]
[函数名称 = 表达式1]
[Exit Function]
[语句段]
[函数名称 = 表达式2]
End Function
```

- Public 是可选参数，表示所有模块的所有其他过程都可访问这个 Function 过程。如果是在包含 Option Private 的模块中使用，则这个过程在该工程外是不可使用的。
- Private 是可选参数，表示只有包含其声明的模块的其他过程可以访问该 Function 过程。
- Friend 是可选参数，只能在类模块中使用。表示该 Function 过程在整个工程中都是可见的，但对于对象实例的控制者是不可见的。
- Static 是可选参数，表示在调用期间将保留 Function 过程的局部变量值。Static 属性对在该 Function 外声明的变量不会产生影响，即使过程中也使用了这些变量。
- <函数名称>指定 Function 过程的名称。
- （参数列表）是可选参数，是在调用时要传递给 Function 过程的参数变量列表。多个变量应用逗号隔开。
- “数据类型”表示 Function 过程的返回值的数据类型，可以是 Byte、 Boolean 、Integer、Long、Currency、Single、Double、Decimal（目前尚不支持）、Date、String（除定长）、Object、Variant 或任何用户定义类型。
- [语句段]中包含 Function 过程中执行的任何语句组。
- 表达式是可选参数，指定 Function 的返回值。

比较 Function 过程和 Sub 过程的定义语法可以看到，两个过程的结构很相似，但是与 Sub 过程不同，Function 过程又有其自身的特点：

（1）在函数第一行的声明语句中，使用“As 数据类型”定义函数的返回值类型。

（2）在函数体内，通过给函数名赋值来指定函数的返回值。

如果函数体内没有“[函数名称 = 表达式 1]”这样的语句，函数将返回一个默认值 0，如果返回数据类型是字符串，则返回一个空字符串。

【例 7.4】手动在代码窗口中创建 Sub 过程，操作过程如下：

01 在 VBE 环境中首先选择【插入】➢【模块】菜单命令，新建一个模块或者直接双击【工程资源管理器】窗口中的某一个模块，打开代码窗口。

02 代码窗口中输入 Function testFunction2 语句，然后按回车键，系统将自动添加后面的括号和 End Function 语句，然后为该函数添加两个参数和返回数据类型，代码如下：

```
Function testFunction2(num1 As Double, num2 As Double) As Double
    testFunction2 = num1 * num2
End Function
```

修改之后 testFunction2 的定义结果如图 7-7 所示。

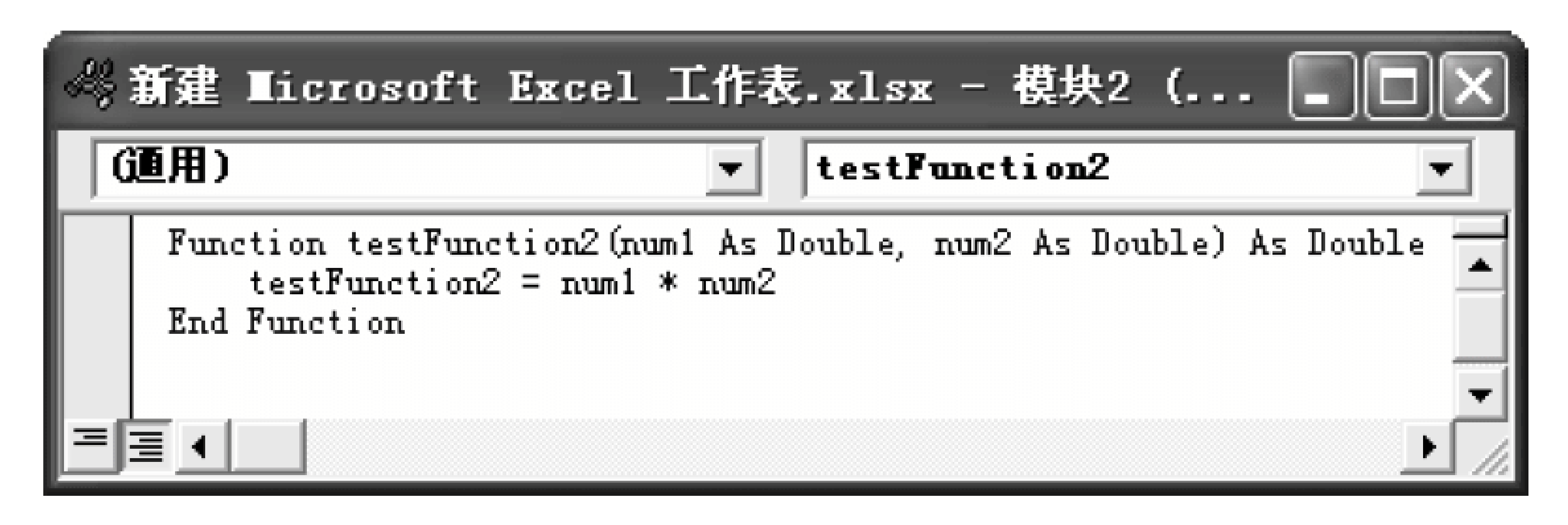


图 7-7　手动创建 Function 过程结构代码



7.4 过程的调用

过程是用来完成某一特定功能的代码模块，程序中可能包含许多这样的过程，只有将这些过程组合在一起才可以实现更大的功能。

7.4.1 调用 Sub 过程

在 VBA 中，Sub 过程的调用分两种形式：一种是在 VBA 代码中直接调用；另一种是在 Excel 中通过调用宏的方式来执行 Sub 过程。

1. 在 VBA 代码中调用

使用代码调用过程又可以分为两种方法。

（1）Call 语句调用，语法格式如下：

```
Call 过程名称[（实际参数）]
```

（2）直接使用过程名调用，即将过程当做一个语句。

【例 7.5】调用过程 TestSub1，操作过程如下。

01 创建 TestSub1 过程。

02 模块代码窗口中输入如下语句，如图 7-8 所示。

```
Sub main()
    Debug.Print "使用 Call 语句调用"
    Call TestSub1

    Debug.Print

    Debug.Print "直接调用"
    TestSub1
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-9 所示。



图 7-8 调用过程

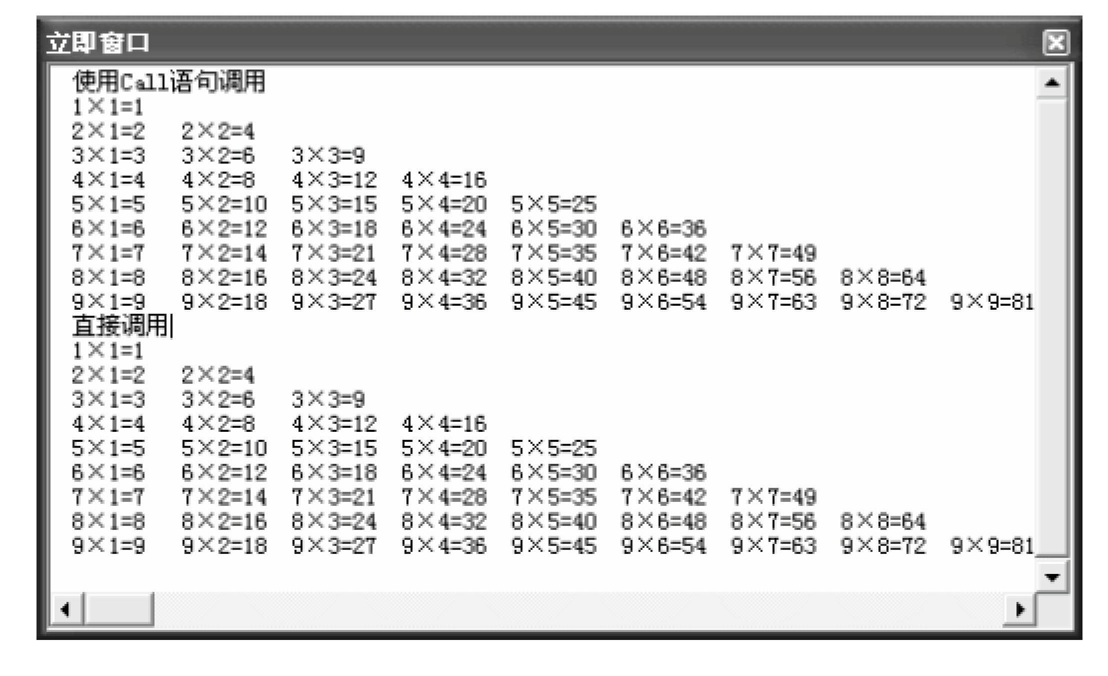


图 7-9 调用过程执行结果

2. 以宏方式调用 Sub 过程

在创建一个 Sub 过程之后，可以在 Excel 工作簿中，选择“录制宏”命令，将 Sub 过程作为一个宏来调用。

【例 7.6】在 Excel 工作簿表单中调用 TestSub1 过程，操作过程如下。

01 返回 Excel 工作簿。

02 在表单中插入一个按钮控件，打开【指定宏】对话框，在【指定宏】对话框中选择宏 TestSub1，如图 7-10 所示。



图 7-10　指定宏对话框

03 单击【确定】按钮，即可完成宏与插入按钮的绑定。

04 单击 Excel 表单中的按钮，之后返回 VBE，可以在【立即窗口】中看到执行的结果，执行结果与图 7-9 所示相同。

7.4.2　调用 Function 过程

Function 过程的调用与 Sub 过程的调用相同，可以在 VBA 代码中使用 Call 语句调用或者直接调用，也可以在 Excel 工作表中调用。

1. 在 VBA 代码中调用

【例 7.7】调用函数 testFunction2，操作过程如下。

01 创建 testFunction2 过程。

02 在模块代码窗口中输入如下语句，如图 7-11 所示。

```
Sub Main()
    Dim num1 As Double, num2 As Double
    Dim numSum As Double
    num1 = InputBox("请输入第 1 个值")
    num2 = InputBox("请输入第 2 个值")
    numSum = testFunction2(num1, num2)
    MsgBox num1 & "乘以" & num2 & "的结果是：" & numSum
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，系统弹出两个输入窗口，这里用户输入两个数值，然后函数 testFunction2 将返回两个数的乘积结果。例如，这里输入值 12.3 和 10，返回结果如图 7-12 所示。




图 7-11　调用 testFunction2 函数



图 7-12　调用函数结果

2. 在 Excel 工作表中调用

自定义的 Function 可以和系统内置函数一样，在 Excel 工作表中进行调用。如果不知道用户自定义的 Function 过程的名称或参数，可以使用【插入函数】对话框帮助用户向工作表中输入这些函数。看下面的例题。

【例 7.8】在 Excel 工作表中调用函数 testFunction2，操作过程如下。

01 返回 Excel 表单窗口，使用鼠标选择一个单元格。

02 在【公式】选项卡的【函数库】中，单击【插入函数】按钮，打开【插入函数】对话框，在【或选择类别】下拉列表中选择【用户定义】选项，下面的函数列表中将显示已有的自定义函数，如图 7-13 所示。

03 在图 7-13 所示的对话框中，单击选择一个自定义函数，单击【确定】按钮，打开【函数参数】对话框，在该对话框中可以输入函数所需要的参数，如图 7-14 所示。



图 7-13　【插入函数】对话框

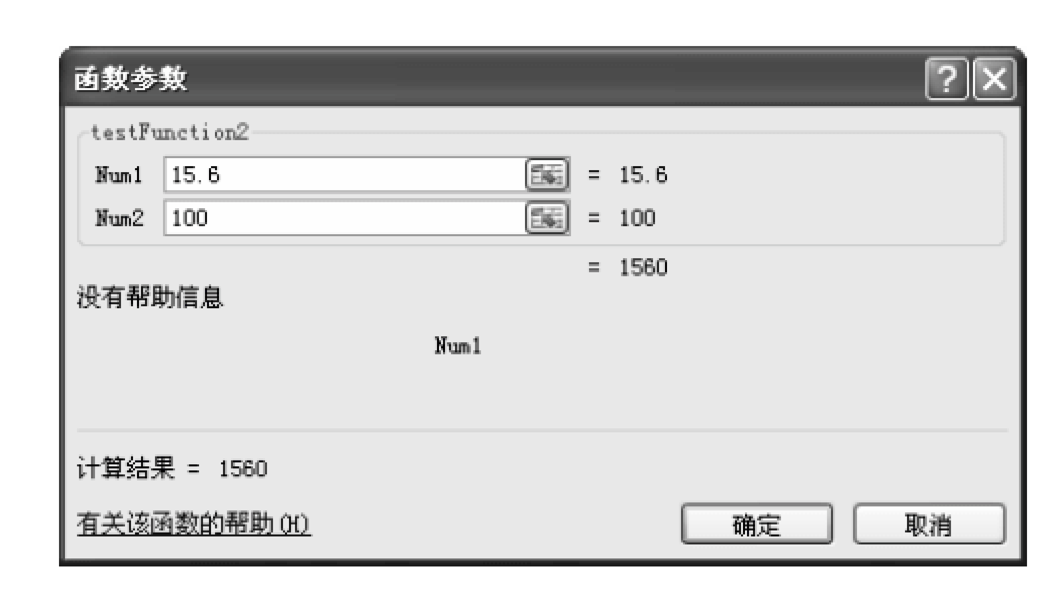


图 7-14　设定函数参数

04 输入完参数后，单击【确定】按钮，完成公式的输入，在 Excel 表单中所选择的单元格里可以看到函数返回的结果，如图 7-15 所示。

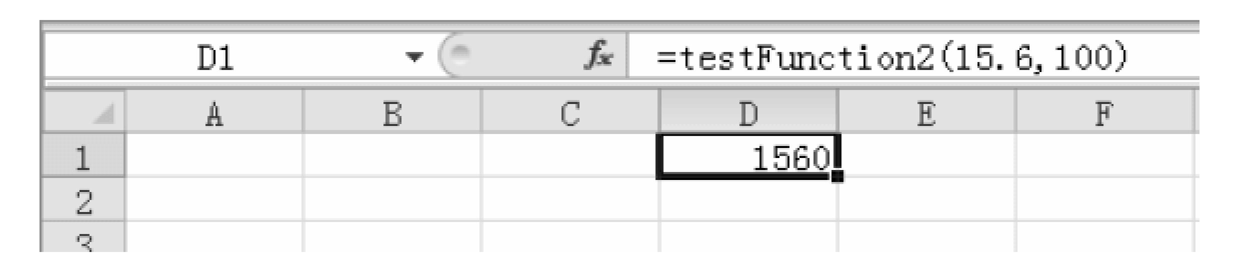


图 7-15　调用 Function 过程结果



7.5 过程的参数传递

过程的参数传递将需要处理的数据交给过程，因为添加了参数，这样就增加了过程的通用性。在 VBA 中有以下一些在过程中传递参数的方式：形参和实参的结合、按值方式传递、按址方式传递和传递数组参数。

7.5.1 形参与实参的结合

形参是过程在定义时出现的参数名称，没有具体的值，例如 testFunction2 过程中：

```
Function testFunction2(num1 As Double, num2 As Double) As Double
    testFunction2 = num1 * num2
End Function
```

这里 num1 和 num2 就是形式参数，在真正调用之前，不知道这两个变量里保存的值是什么。

实参是实际调用时，传递给过程的参数的具体值，这些具体的值可以是常量、变量或者数组等。

VBA 中，形参与实参结合时有两种方式：按位置结合和按参数名称结合。

1. 按位置结合

这种结合方式表示在调用过程时，传递的参数的顺序必须与定义过程时参数的顺序完全相同。

【例 7.9】使用位置结合的方式调用带参数的 Sub 过程，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 7-16 所示。

```
Sub Main()
   computeVolume 3, 4, 5, "长方体"
End Sub

Public Sub computeVolume(length As Double, _
                    width As Double, _
                    height As Double, _
                    name As String)
    Dim volume As Double
    volume = length * width * height
    Debug.Print name & "的体积是： " & volume
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-17 所示。

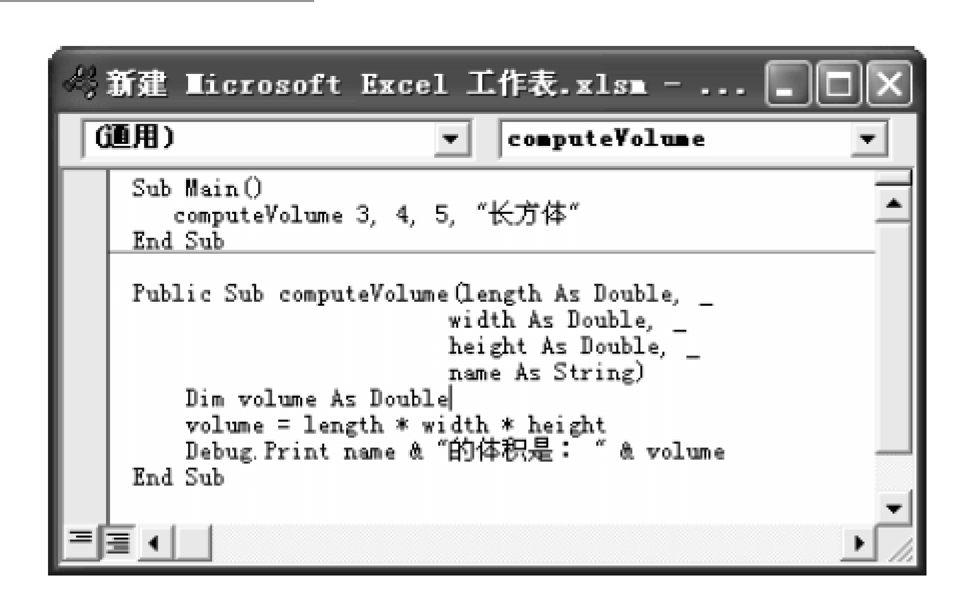


图 7-16 按位置结合方式调用过程

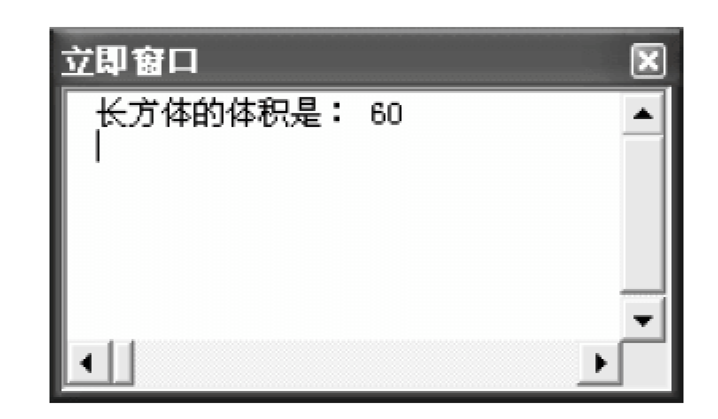


图 7-17 调用 computeVolume 的结果

【提示】由模块中的代码可以看到，这里在调用 computeVolume 时，传递的三个参数的类型必须与过程定义时的顺序相同，在 Main 过程中调用 computeVolume 时，3 赋值给 length，4 赋值给 width，5 赋值给 height，“长方体”赋值给 name 参数。

2. 按参数名称结合

使用按参数名称结合的方式调用时，输入形参的名称，将形参的名称与实参用“:=”符号连接起来，另外，使用参数名称结合时不需要按照过程定义时的顺序给出参数，参数位置可以随意放置。

【例 7.10】使用参数名称结合的方式调用带参数的 Sub 过程，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 7-18 所示。

```
Sub Main()
  computeVolume height:=5, length:=3, name:="长方体", width:=4
End Sub

Public Sub computeVolume(length As Double,
                        width As Double,
                        height As Double, _
                        name As String)
    Dim volume As Double
    volume = length * width * height
    Debug.Print name & "的体积是：  " & volume
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，这里的结果和上一个例题中的完全相同，如图 7-17 所示。

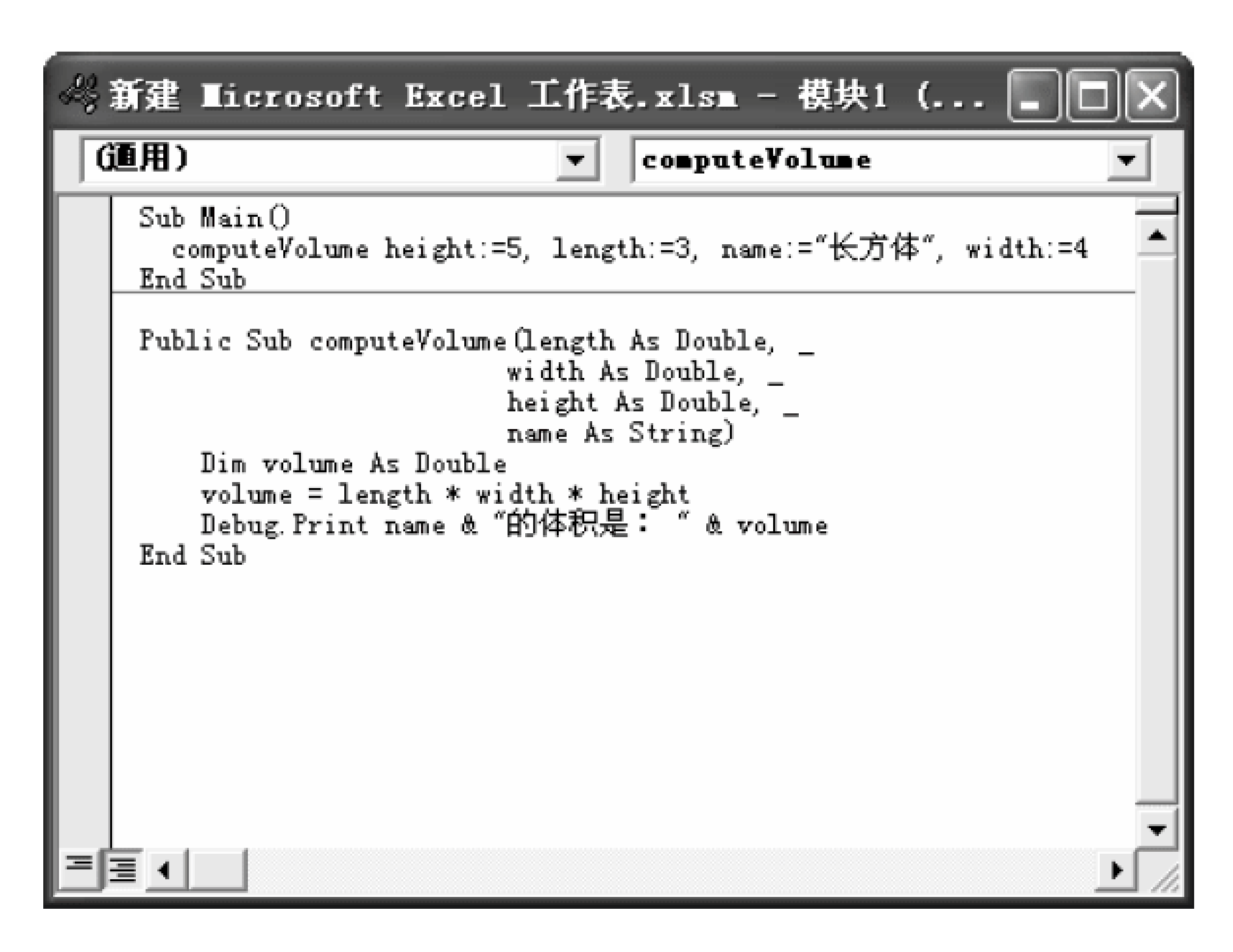


图 7-18 使用参数名结合方式调用带参数的子过程

这两种方法实现了相同的效果，虽然第二种按名称传递参数时需要指定形参的名称，但是使用这种方式将增强程序的可读性，降低程序出错的可能性。

7.5.2 按传值方式传递参数

以传值的方式调用过程时，传递给过程的实际是值的一个副本，定义过程时，在需要传值的变量前面添加 ByVal 关键字，则该参数就按照传值的方式传递到过程中。因为传递的是一个变量的副本，所以在过程中对参数的任何变动都不会影响到外部调用程序中的变量的值。这与下一小节将要介绍的传址的方式不同。

【例 7.11】使用按值传递的方式调用过程，操作过程如下：

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 7-19 所示。

```
Sub Main()
    Dim x As Integer
    x = 25
    Debug.Print "主过程中变量 x = " & x
    testValue x
    Debug.Print "主过程中变量 x = " & x
End Sub

Public Sub testValue(ByVal x As Integer)
    x = x + 10
    Debug.Print "子过程中变量 x = " & x
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-20 所示。

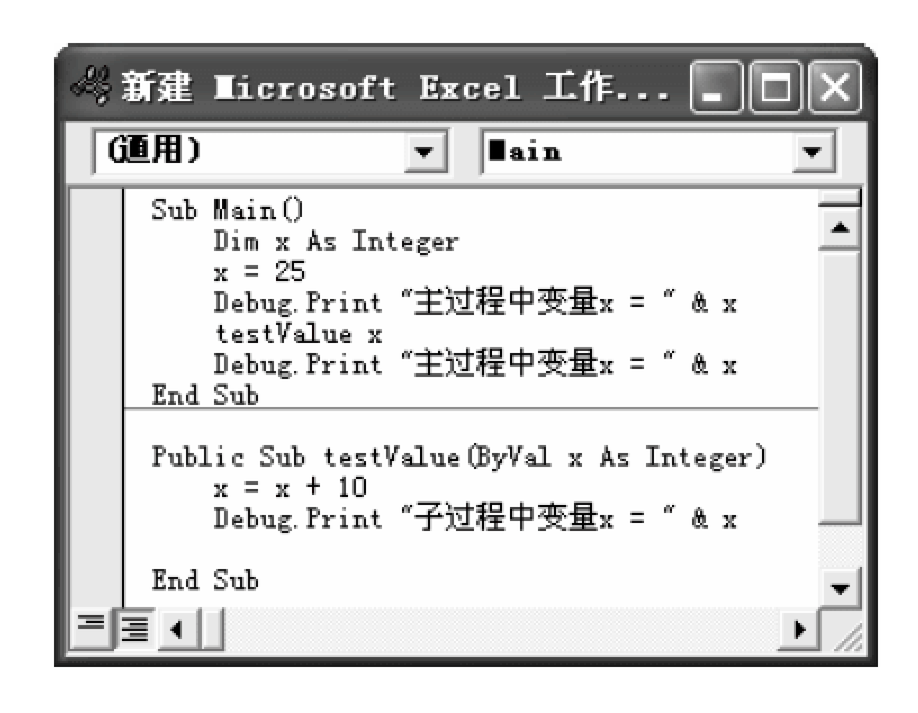


图 7-19　按值传递方式调用过程

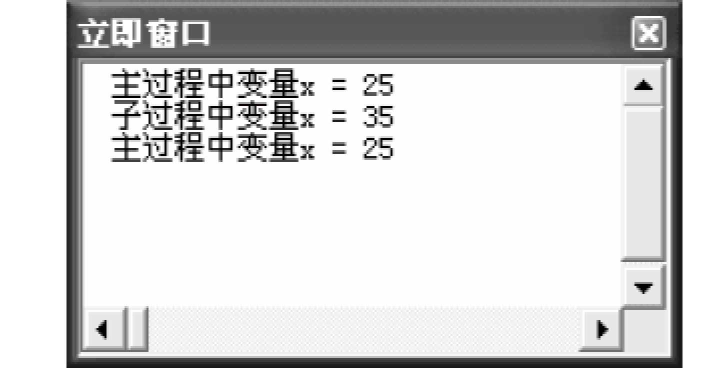


图 7-20　testValue 调用结果

由图 7-20 打印输出的结果可以看到，在调用传值过程前 x 变量的值是 25，在 testValue 过程中对 x 变量进行了修改，打印结果为 35，但是返回到 Main 过程中之后，x 的值并没有真的改变，仍然是 25，这样就保证了子过程不能随意修改调用程序中的变量。

7.5.3　按传地址方式传递参数

在 VBA 中另一种传递实参的方法是传地址，即传递变量在内存中的地址，而不是变量本身的值，定义过程时，在需要传值的变量前面添加 ByRef 关键字。传地址是 VBA 中默认的方式，在定义过程时，如果没有在形参前面添加 ByVal 或 ByRef，则默认都是通过传地址方式传递。因为传递的是实参的地址，所以原调用过程可以看到被修改的实参的值。

【例 7.12】使用按地址传递的方式调用过程，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 7-21 所示。

```
Sub Main()
    Dim x As Integer
    x = 25
    Debug.Print "主过程中变量 x = " & x
    testRef x
    Debug.Print "主过程中变量 x = " & x
End Sub

Public Sub testRef(ByRef x As Integer)
    x = x + 10
    Debug.Print "子过程中变量 x = " & x
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-22 所示。

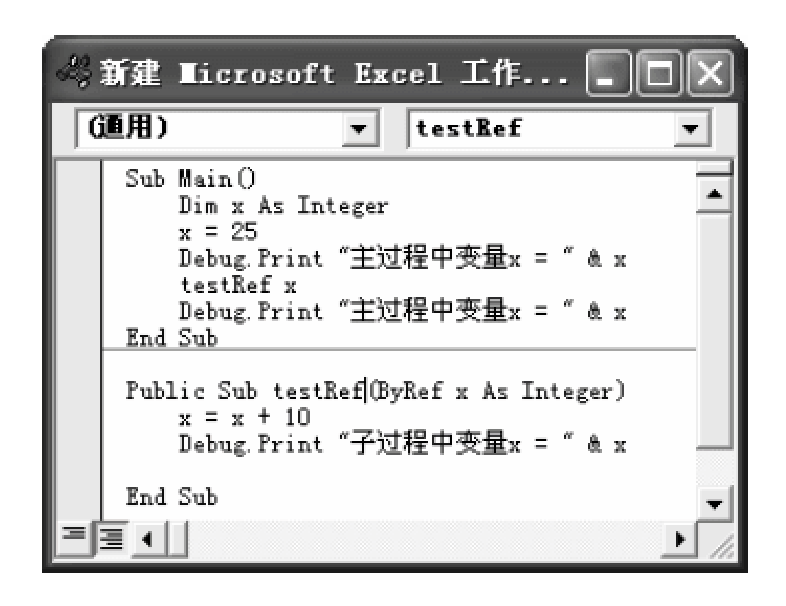



图 7-21　按地址传递调用过程

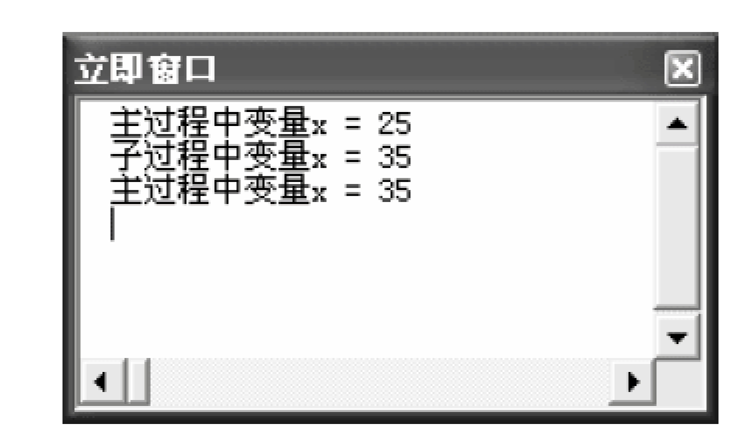


图 7-22　testRef 过程调用结果

由打印输出的结果可以看到，在调用传址过程前 x 变量的值是 25，在 testRef 过程中对 x 变量进行了修改，打印结果为 35，返回到 Main 过程中之后，因为形参中的 x 和 Main 过程中传递给 testRef 过程的实参 x 指向同一块内存单元，所以返回调用过程后，实参 x 的值也发生了变化。

【技巧】Sub 过程不能返回运算结果，但是如果需要 Sub 过程返回值，可以使用 ByRef 方式定义形参就可以将子过程的运算数据返回到调用程序中。

【提示】程序中使用传地址比传值效率高，但是传地址方式中的形参并不是一个真正的局部变量，有可能对程序产生不必要的影响。如果没有特殊要求，应尽量使用传值方式。

7.5.4　传递数组参数

除了上面的两种传递参数的方法之外，VBA 中还允许将数组作为参数传递到过程中。在定义过程时，数组作为形参出现在参数列表中，形参数组只能按地址传递参数，对应的实参也必须是数组，且数据类型相同。调用过程时，把要传递的数组名放在实参表中，数组名后面不跟圆括号。在过程中不可以用 Dim 语句对形参数组进行声明，否则会产生“重复声明”的错误。

在使用传递数组参数时要注意以下几点。

（1）因为数组是内存中的一片连续区域，数组参数只能按地址传递。

（2）在声明数组为形参时，必须要包含括号，以便与一般变量区分，但可以不声明大小。

（3）在实参和形参中不需要说明数组的维数。

（4）在遍历数组中的元素时，可以使用 LBound 和 UBound 函数获取数组的上界和下界。

（5）因为是按地址传递，所以过程中对形参数组元素的改变，将最终改变调用过程中实参数组中相应元素的值。

【例 7.13】使用过程求出指定数组中的最小值，操作过程如下：

01 打开 VBE，依次选择【插入】➢【模块】菜单命令。

02 在模块中输入以下代码，如图 7-23 所示。

```
Sub Main()
```




```
    Dim myArray(1 To 10) As Integer
    '向数组中插入 10 个 0~100 之间的随机数
    For idx = 1 To 10
        myArray(idx) = Rnd * 100
        Debug.Print myArray(idx);
    Next
    Debug.Print '打印一个空白行
    Call getMin(myArray)
End Sub

Public Sub getMin(myArray() As Integer)
    Dim idx As Integer, min As Integer
    min = myArray(LBound(myArray))
    For idx = LBound(myArray) To UBound(myArray)
        If myArray(idx) < min Then min = myArray(idx)
    Next idx
    Debug.Print "数组中最小值等于 " & min

End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-24 所示。



图 7-23 调用数组参数的过程

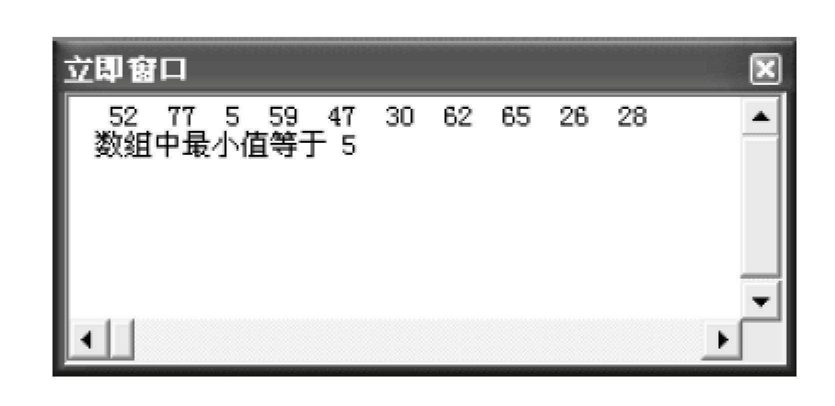


图 7-24 getMin 过程执行结果

getMin 过程中传递了一个整数类型的数组，首先将数组中第一个值保存在 min 变量中，然后依次让数组中每一个值与 min 值比较，如果数组元素比 min 小，则将其赋值给 min，这样比较完之后，min 变量中始终保存着数组中最大的数。



7.6 可选参数和可变参数

前面介绍的 Sub 过程和 Function 过程中，参数的个数都是固定的，并且在调用时必须要指定相应的实参。VBA 中可以根据需要设置可选参数和可变参数。

7.6.1 可选参数

可选参数即在函数或过程定义时的形参是固定的，但在调用时可以不用指定某个实参，如在定义时有 3 个实参，调用时可以只指定前 2 个实参。VBA 中使用 Optional 关键字来设置该形参为可选参数。在过程内部可以使用 IsMissing 函数来检测调用程序是否传递了该可选参数。定义可选参数的语法格式如下：

Sub/Function 过程名称（…Optional 变量名）

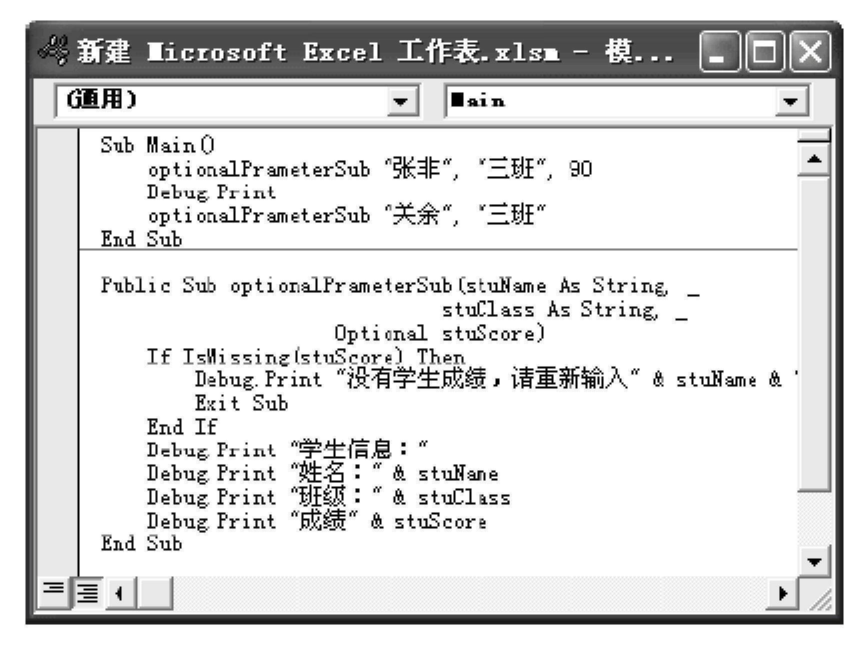
【例 7.14】定义可选参数，使用并判断是否传递了可选参数，操作过程如下：

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 7-25 所示。

```
Sub Main()
    optionalPrameterSub "张非", "三班", 90
    Debug.Print
    optionalPrameterSub "关余", "三班"
End Sub
Public Sub optionalPrameterSub  (stuName As String, _
                                stuClass As String, _
                                Optional stuScore)
    If IsMissing(stuScore) Then
        Debug.Print "没有学生成绩，请重新输入" & stuName & "的成绩"
        Exit Sub
    End If
    Debug.Print "学生信息："
    Debug.Print "姓名：" & stuName
    Debug.Print "班级：" & stuClass
    Debug.Print "成绩" & stuScore
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-26 所示。



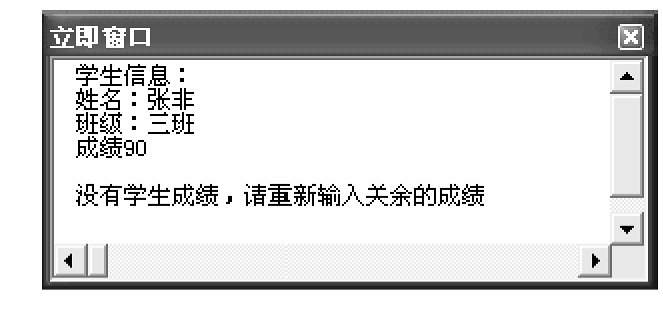




图 7-25　定义可选参数过程　　图 7-26　optionalPrameterSub 过程执行结果

该段代码中定义过程 optionalPrameterSub 中第 3 个参数是一个可选形参，在 Main 过程中，调用这个过程时，可以提供三个参数，也可以提供两个参数。可以看到使用 IsMissing 可以判断是否指定了第 3 个参数。

【提示】在过程中可以定义多个可选参数，但可选参数必须放在参数列表的最后，而且必须是 Variant 类型，如果可选参数也指定了一个数据类型，则 IsMissing 函数将无法对其进行判断。

7.6.2 可变参数

可选参数表示是否选择使用，而可变参数则表示过程中实参的个数是不确定的。定义可变参数时，在最后一个参数前面加上 ParamArray 关键字，之后过程将可以接受任意个数的参数。

【例 7.15】使用可变参数过程编写求和函数，操作过程如下。

01 打开 VBE，依次选择【插入】➤【模块】菜单命令。

02 在模块中输入以下代码，如图 7-27 所示。

```
Sub Main()
    Dim sum1 As Integer, sum2 As Integer
    getSum sum1, 1, 3, 4, 6, 8, 12, 53, 5
    Debug.Print " sum1 等于 : " & sum1
    getSum sum2, 2,5,12,12
    Debug.Print " sum2 等于 : " & sum2
End Sub

Public Sub getSum(sum As Integer, ParamArray intNum())
    For idx = LBound(intNum) To UBound(intNum)
        sum = sum + intNum(idx)
    Next
End Sub
```

03 单击工具栏中的【运行子过程/用户窗体】按钮，结果输出在【立即窗口】中，如图 7-28 所示。

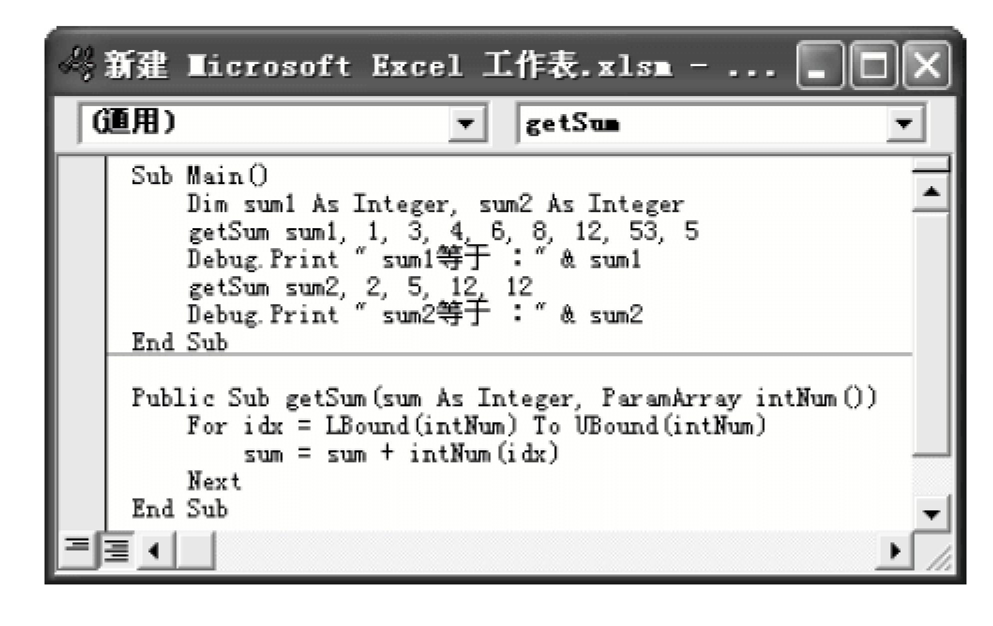



图 7-27　创建可变参数过程

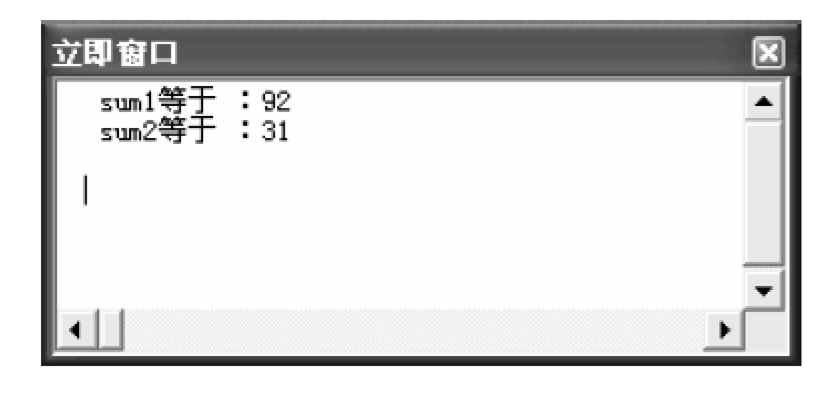


图 7-28　getSum 过程执行结果

可以看到，getSum 中定义两个形参，第一个是按地址传递的形参，用于返回计算之后的结果；第二个形参使用 ParamArray 将其定义为可变参数，可变参数是一个数组，程序中使用 LBound 和 UBound 函数获取数组的长度，然后对数组中的值进行累加，最后使用 sum 返回结果。

【提示】ParamArray 关键字只能用于参数列表中的最后一个参数，同时该关键字不能与 ByVal，ByRef 或 Optional 一起使用。



7.8 综合实战——彩票号码生产代码

相信有很多读者都买过彩票，中彩票是有一定概率的，这里为读者介绍一段简短的代码来生产彩票中奖号码的，创建完成之后，赶紧试试自己的手气吧！参考代码如下。

```
Public Sub Lottery(lott() As Integer)
   ' Dim lott(1 To 10) As Integer, i As Integer
    Randomize
    For i = 1 To 10
        lott(i) = Int(Rnd() * 10)
    Next i
    'Lottery = lott
End Sub
Sub main()
    Dim arr(1 To 10) As Integer
    Lottery arr
    Debug.Print "本期的中奖号码是:"
    For i = 1 To 10
        Debug.Print arr(i) & " ";
    Next
    Debug.Print
End Sub
```



7.9 高手私房菜

技巧 1：为什么 Function 过程不出现在宏对话框中?

与 Sub 过程不同，自定义函数 Function 过程并不出现在宏对话框中；当执行 VBE 编辑器中的【运行】➤【运行子过程/用户窗体】命令时，如果光标位于某 Function 过程中，就不能获取宏对话框并从中选择要运行的宏。因此，在开发过程的时候，必须采取其他方式对自定义函数进行测试，可以设置调用该函数的过程；如果该函数是用在工作表公式中的，可以在工作表中输入简单的公式进行测试。

技巧 2：指定为 ParamArray 的参数的数据类型必须相同吗?

ParamArray 参数的每个参数可以是不同的数据类型。然而，参数本身必须声明为一个 Variant 类型的数组。可以提供任何数量的参数给 ParamArray。当调用发生时，调用所提供的每个参数，都会变成 Variant 数组中一个相对应的元素。

第 8 章　Excel 工作簿相关操作

通过本章的学习，可以使读者了解工作簿的操作方法和属性，实现 Excel 工作簿对象的新建、打开、保存及关闭等操作。如新建仅包含一张工作表的工作簿、新建工作簿到指定位置、判断一个工作簿是否已打开、保存对工作簿的更改、关闭并保存所有工作簿等。



8.1 新建工作簿

当需要利用 VBA 完成一个新任务时，会涉及 Excel 工作簿的新建问题，本节将详细介绍几种新建工作簿的方法。

8.1.1 新建空白工作簿

使用 Excel 新建空白工作簿的方法很多，最常见的是通过选择【文件】➢【新建】菜单命令。这里要介绍的是使用 Excel VBA 代码新建空白工作簿的方法，具体的操作步骤如下。

01 在 Excel 中，选择【开发工具】选项卡，在【代码】选项组中单击 Visual Basic 按钮，打开 VBA 代码窗口，选择【插入】➢【模块】菜单命令，并输入如下代码：

```
Sub 新建工作簿()
Workbooks.Add      '利用工作簿集合的 Add 方法来添加新工作簿。
End Sub
```

02 运行上述代码即可成功新建一个空白工作簿。

【提示】在上述代码中，Workbooks 代表工作簿集合对象，Workbook 代表工作簿对象。创建工作簿必须使用 Workbooks，如果使用 Workbook.Add 代码创建工作簿，运行后会弹出一个错误提示对话框，如图 8-1 所示。



图 8-1　错误提示对话框

8.1.2　控制工作簿中工作表的个数

在默认情况下新建的工作簿中包含 3 张工作表。新建工作簿中工作表的数目可通过 Excel 中的【Excel 选项】对话框进行设置，即在 Excel 中选择【文件】➤【选项】菜单命令，打开【Excel 选项】对话框，在【包含的工作表数】微调框中进行相应设置即可，如图 8-2 所示。

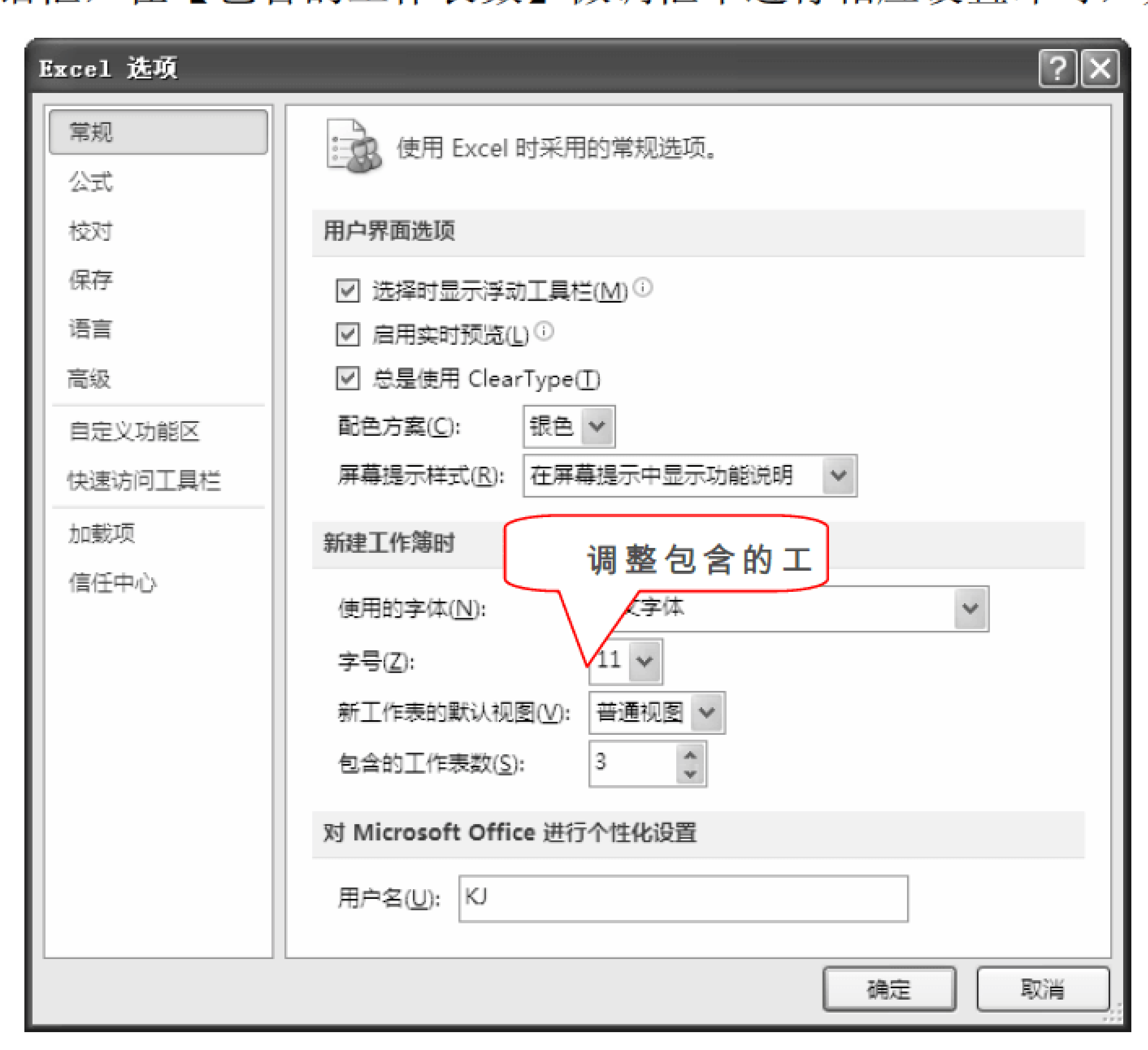


图 8-2　【Excel 选项】对话框

在 VBA 中 Workbooks 集合的 Add 方法的语法格式为如下。

```
Add(Template)
```

其中的 Template 为可选参数，用于确定如何创建工作簿。如果将该参数指定为 xlWBATemplate 常量中的一种，则新建工作簿中将会隐式指定类型的单张工作表。

关于 xlWBATemplate 常量说明如表 8-1 所示。

表 8-1　xlWBATemplate 常量说明

xlWBATemplate 常量	说明
xlWBATWorksheet	工作表
xlWBATChart	图表
xlWBATExcel4MacroSheet	宏表
xlWBATExcel4IntlMacroSheet	国际通用宏表

新建仅包含一张工作表的工作簿具体操作步骤如下。

01 在 VBA 代码窗口中输入如下代码，如图 8-3 所示。

```
Sub 新建工作簿()
Workbooks.Add (xlWBATWorksheet)
End Sub
```

02 运行上述代码后，即可成功新建一个仅包含一张工作表的工作簿，如图 8-4 所示。

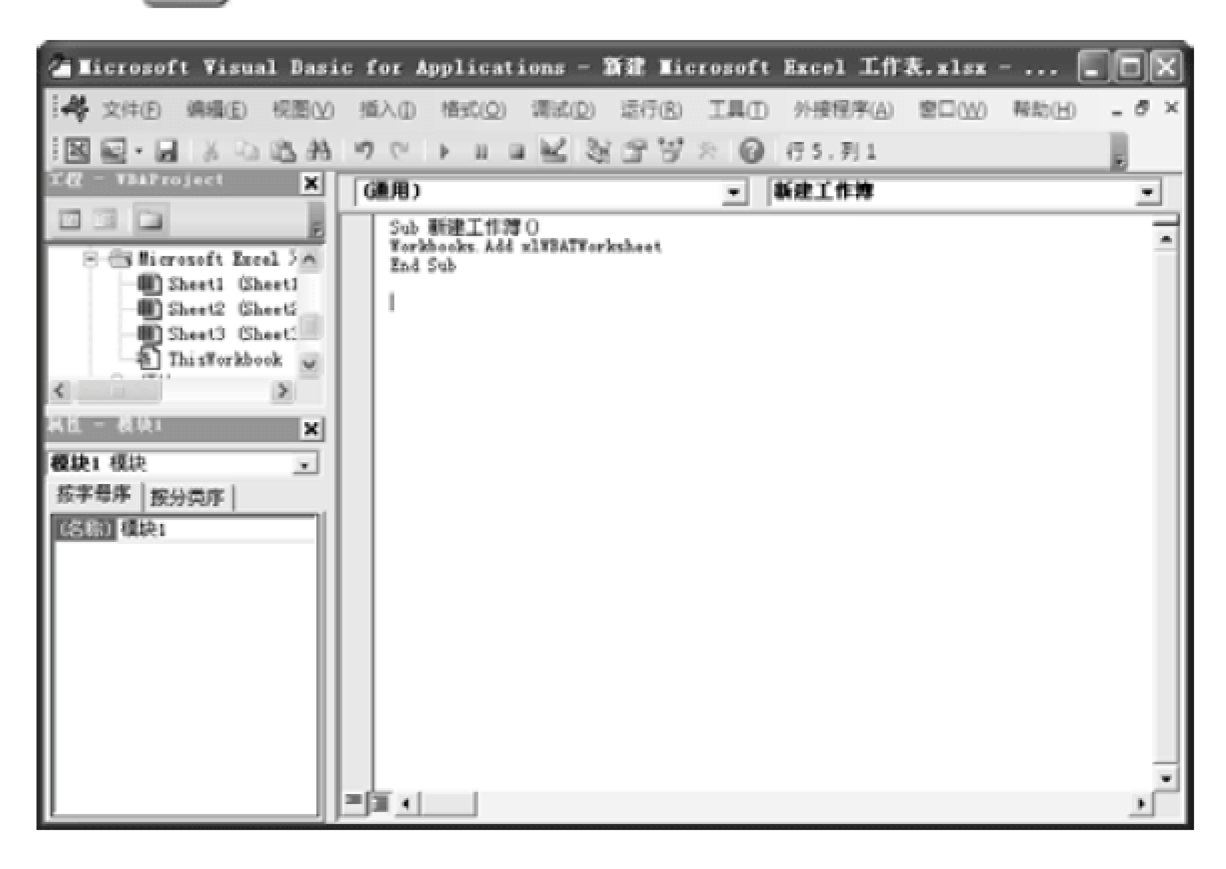


图 8-3　输入代码

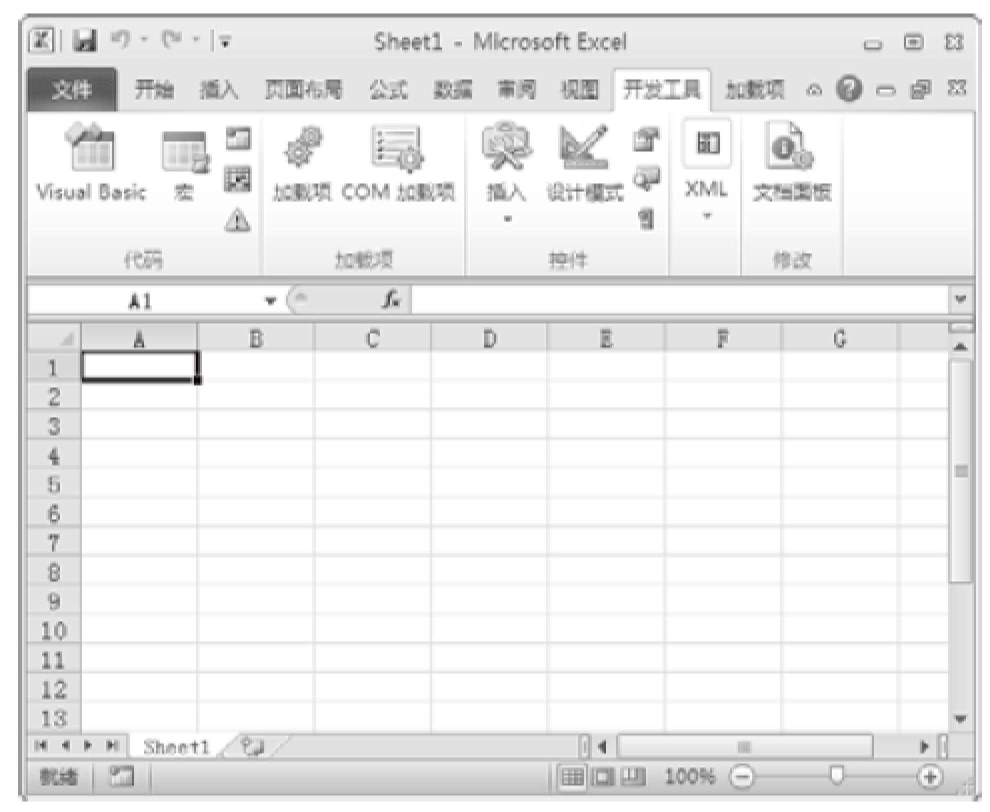


图 8-4　新建仅包含一张工作表的工作簿

8.1.3 创建自定义属性的工作簿

8.1.1 节中创建工作簿的方法比较简单，但是在实际的工作中存在较大的缺陷。因为用户不能设置工作簿的属性，从而无法利用新建工作簿完成指定的操作。

下面将来讲述如何创建通过分配变量，来创建自定义属性的工作簿。

本实例主要讲述如何创建“英达图书销售工作簿”，具体操作步骤如下。

01 在 Excel 中，选择【开发工具】选项卡，在【代码】选项组中单击【Visual Basic】按钮，打开 VBA 代码窗口，选择【插入】➢【模块】菜单命令，并输入如下代码。

```
Sub 新建图书销售工作簿()
Dim MyBook As Workbook
```

```
Set MyBook = Workbooks.Add
With MyBook
.Title = "图书销售工作簿"
.SaveAs Filename:=" 英达图书销售工作簿"
.AddToFavorites
End With
End Sub
```

02 运行代码后，即可成功创建“英达图书销售工作簿”，位置为默认保存到“我的文档”，效果如图 8-5 所示。

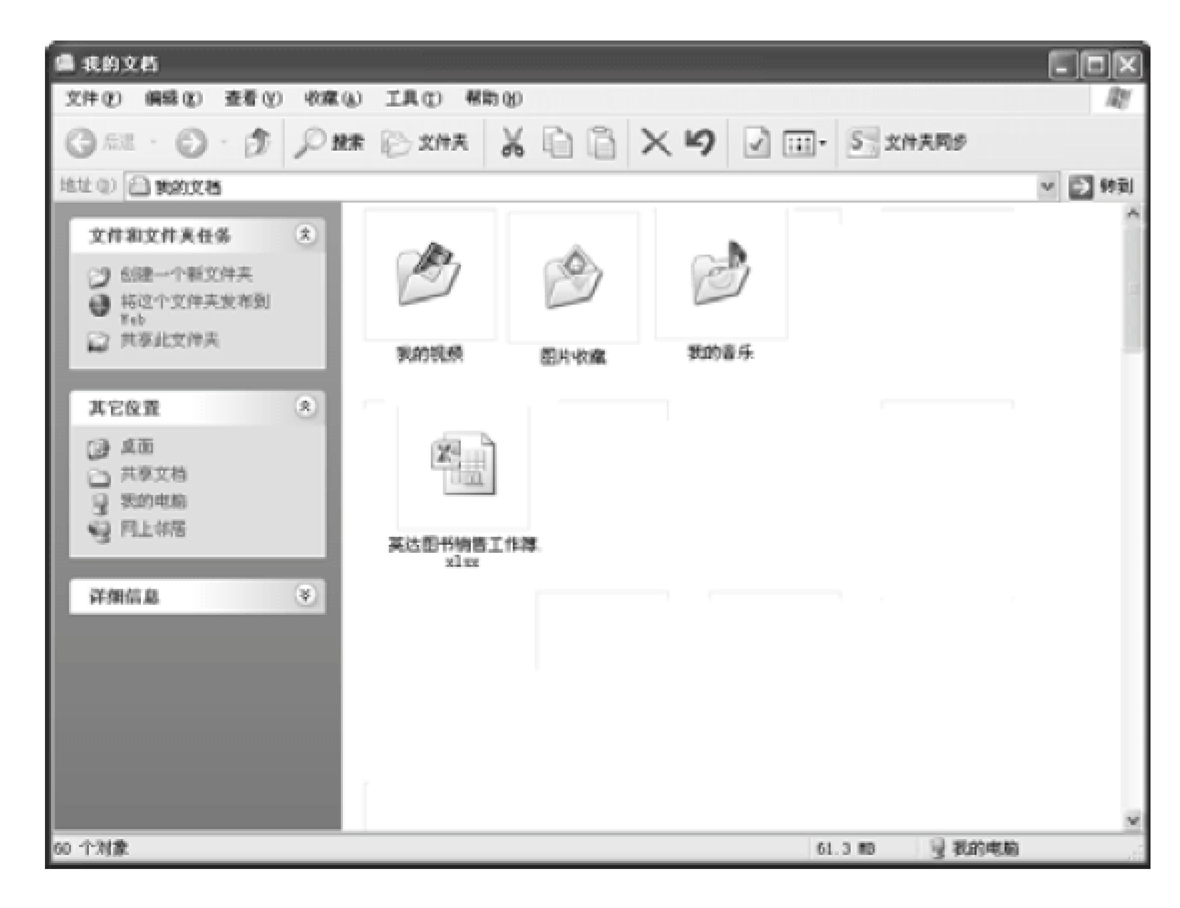


图 8-5　新建工作簿的位置和名称

上述代码分析如下。

- With MyBook.Title = “图书销售工作簿”语句定义工作簿保存为 Web 页时的标题。
- SaveAs Filename:= “英达图书销售工作簿”语句定义工作簿的名称。
- AddToFavorites 语句主要是控制保存位置为默认位置（我的文档）。如果用户想自定义保存位置，可以参照下一节的内容。



8.2 保存工作簿

工作簿在新建和编辑后，需要对其进行保存操作。保存工作簿分为保存工作簿和另存为其他工作簿。

8.2.1 保存工作簿的两种情况

在 Excel 中，可以利用 VBA 保存对工作簿的更改，主要分为保存修改后的单个文件和同时保存所有已打开的文件。

1. 保存单个工作簿

如果只对指定名称的工作簿进行保存，保存的基本语法如下。

```
Workbooks("工作簿名称").Save
```

如果要保存当前活动的工作簿，可以直接使用如下语句。

```
Activeworkbook.Save
```

2. 保存所有打开的工作簿

当同时打开多个工作簿文件时，想要实现同时保存所有的工作簿，则可使用下面的程序。

```
Sub 保存所有工作簿()
Dim i As Workbook          '声明变量 i 为工作簿对象，该变量用于后面的循环
For Each i In Workbooks     '变量 i 遍历 Workbooks 集合对象中的所有工作簿
    If i.Path <> "" Then    '对工作簿的 Path 属性进行判断，判断工作簿的路径是否为空
      i.Save               '将每一个 Path 属性不为空的文件按照原来的存储路径进行保存
    End If
Next i
End Sub
```

运行上述代码，则可以将所有已存在的文件进行保存。

8.2.2 将工作簿进行另存

在日常应用中有时需要将已有的工作簿另存为其他文件，或者需要将工作簿备份。在 Excel VBA 中可以完成这些工作，具体操作如下。

1. 另存为其他工作簿

另存为其他工作簿的代码格式如下。

```
Workbooks("工作簿名称").SaveAs 另存的完整路径和文件名
```

把当前文件关闭并把该文件另存到 E 盘，文件名为 Myworkbook.xlsx，具体操作命令如下。

```
Sub 另存为()
ActiveWorkbook.SaveAs "E:\ Myworkbook.xlsx"
End Sub
```

这里的 SaveAs 可实现“另存为”操作。

2. 备份工作簿

备份工作簿是指在不关闭当前文件时将文件复制到指定的路径下。按照指定路径备份工作簿的代码格式如下：

```
Workbooks("工作簿名称").SaveCopyAs 备份的完整路径和文件名
```

把当前文件备份到 E 盘，文件名为 workbook1.xlsx，具体操作命令如下。

```
Sub 备份文件()
ActiveWorkbook.SaveCopyAs "E:\workbook1.xlsx"
End Sub
```

运行这段代码，则可将当前文件备份到 E 盘中，并且当前文件不关闭。

8.2.3 为工作簿添加密码

为了安全起见，可以为工作簿添加打开密码，具体操作命令如下。

```
Workbooks("工作簿名称").SaveAs 完整路径和文件名, Password:="设置的密码"
```

例如，下面将当前文件另存到 E 盘，文件名为“Myworkbook”，设置打开密码为“abc123”，具体操作步骤如下。

01 在代码窗口中输入如下代码，如图 8-6 所示。

```
Sub 另存并加密()
ActiveWorkbook.SaveAs "E:\Myworkbook.xlsx", Password:="abc123"
End Sub
```

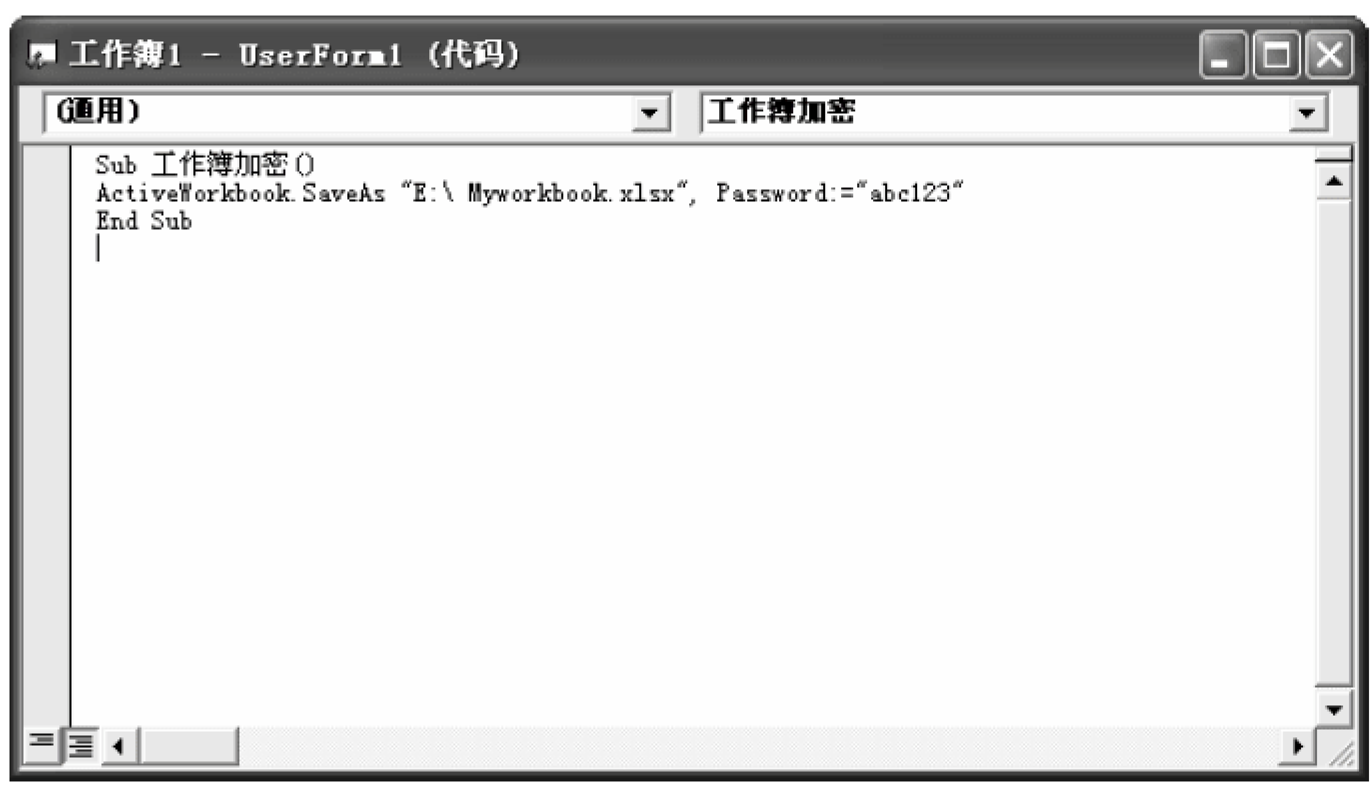


图 8-6　输入“另存并加密”功能的程序

02 运行该程序后，即可在指定位置处保存当前文件，并为其添加密码。当再次打开该文件时，即可弹出一个【密码】对话框，必须输入正确密码后，才可打开该文件，如图 8-7 所示。



图 8-7　【密码】对话框



8.3 打开工作簿

要继续未完成的 VBA 任务时，需要执行工作簿的打开操作，本节将详细介绍打开工作簿的几种操作方法。

8.3.1 打开指定工作簿

打开已有工作簿的方法有多种，如在硬盘中打开文件的目录，双击已存的 Excel 文件，或通过选择【文件】➢【打开】菜单命令，也可以通过单击常用工具栏中的【打开】按钮打开指定工作簿。以上操作相对简单，本节主要介绍通过 Excel VBA 代码来实现工作簿的打开操作。

由于每个工作簿都有相对应的路径，而且在硬盘中其他位置还可能存在有同样文件名的 Excel 文件，所以在打开时要指定路径。

打开指定工作簿的代码如下。

```
Workbooks.Open Filename: = 文件名和路径
```

【提示】Filename:=可以省略；Open 后要有一个半角空格。

例如，打开 E 盘中的 myworkbook.xlsx 文件，具体的操作步骤如下。

01 在 VBA 代码窗口中，插入一个新模块，并输入如下代码。

```
Sub 打开工作簿()
    Workbooks.Open Filename:="E:\ myworkbook.xlsx"
End Sub
```

02 运行上述代码即可打开相应的文件。

要打开指定文件，也可以使用以下代码。

```
Sub 打开工作簿()
    Workbooks.Open "E:\ myworkbook.xlsx"
End Sub
```

【提示】如果所指定的文件不存在，会弹出如图 8-8 所示的错误提示信息对话框。

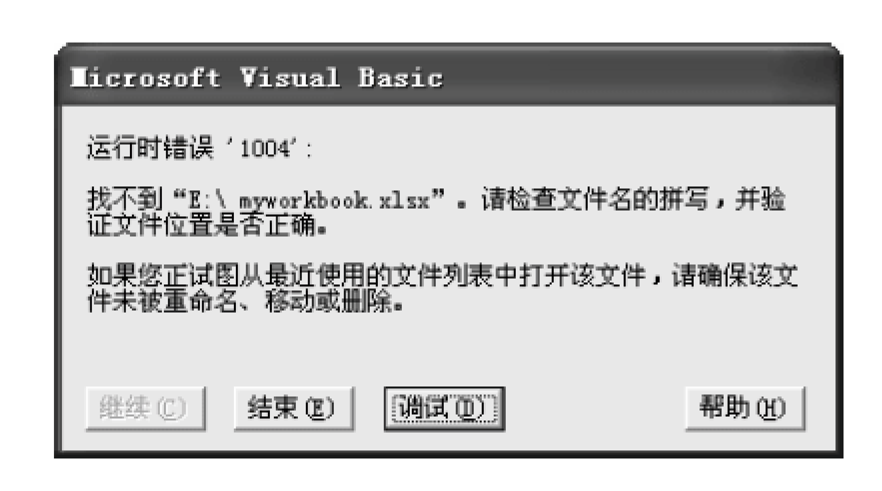


图 8-8　错误提示信息

8.3.2　打开设置保护的工作簿

为了安全需要为工作簿设置安全保护，在 Excel 中设置打开权限密码保护的具体操作步骤如下。

01 在 Excel 工作簿中，选择【文件】➤【另存为】菜单命令，即可打开【另存为】对话框，在【保存位置】下拉列表中选择文件的存放位置，然后单击【工具】下三角按钮，在其下拉菜单中选择【常规选项】，如图 8-9 所示。

02 即可打开【常规选项】对话框，在【打开权限密码】文本框中输入要设置的密码“111111”，然后单击【确定】按钮，如图 8-10 所示。



图 8-9　【另存为】对话框　　　　图 8-10　【常规选项】对话框

03 即可打开【确认密码】对话框，在【重新输入密码】文本框中输入前面设置的密码"111111"如图 8-11 所示。

04 单击【确定】按钮返回【另存为】对话框，然后单击【保存】按钮即可。

对于已设置密码保护的工作簿，如果用“Workbooks.Open+工作簿地址”的方法打开，系统会弹出一个【密码】对话框，提示用户输入打开密码，如图 8-12 所示。



图 8-11 【确认密码】对话框　　　图 8-12 提示输入打开密码

为了方便，用户可以在不出现密码输入提示框的情况下打开文件，代码格式如下。

```
Workbooks.Open Filename:=文件名和路径, Password:=文件打开密码
```

打开 E 盘中文件名为 myworkbook.xlsx 且密码为 abc123 的文件，具体操作步骤如下。

05 在 VBA 代码窗口中输入如下代码。

```
Sub 打开密码保护文件()
Workbooks.Open Filename:="E:\ myworkbook.xlsx", Password:="abc123"
End Sub
```

06 运行上述代码后，系统不再弹出【密码】对话框，直接将文件打开。

8.3.3 判断某个工作簿是否已打开

已经打开了多个工作簿，要判断哪个工作簿是否已经打开时，手工一个个地验证会很麻烦，如果使用 VBA 代码判断会比较简单，具体操作方法如下。

例如，判断文件名为 workbook1.xlsm 的工作簿是否已打开，具体操作命令如下。

```
Sub 判断文件打开状态()
Dim i As Integer          '声明变量 i 为整数，变量应用于后面的循环
For i = 1 To Workbooks.Count  '设置变量 i 在 1 到“工作簿数量”的区间内进行循环
   If Workbooks(i).Name = "workbook1.xlsm" Then
'把第 i 个工作簿的文件名和给定的“workbook1.xlsm”相对比，如果相同就执行下一句
       MsgBox "文件已打开"       '用对话框显示“文件已打开”
       Exit Sub                 '跳出当前过程
   End If
Next i
MsgBox "文件未打开"
'如果程序没有被中止，就说明找不到相同的文件名，该文件未被打开，而如果找到了相同的文件名，
'程序会在运行该句之前中止该过程
End Sub
```

运行上述代码，如果该文件已打开，系统即可弹出一个如图 8-13 所示的提示信息；如果该文件未打开，则弹出如图 8-14 所示的提示信息。



图 8-13　文件已打开提示信息



图 8-14　文件未打开提示信息



8.4 关闭工作簿

工作簿操作完成并保存后，需要将其关闭。关闭工作簿包括关闭指定名称的工作簿和关闭所有工作簿。

8.4.1 关闭指定名称的工作簿

当打开多个工作簿时，可以使用下面的代码关闭指定名称的工作簿。

```
Workbooks("工作簿名称").Close
```

例如，关闭已打开的多个工作簿中的名为“Myworkbook.xlsx”的工作簿，具体 VBA 代码如下。

```
Sub 关闭指定工作簿()
Workbooks("Myworkbook.xlsx").Close
End Sub
```

【提示】当指定的工作簿不是打开状态时，在运行上述程序后会出现错误提示信息，如图 8-15 所示。



图 8-15　错误提示信息

8.4.2 关闭所有工作簿

可以使用 Workbooks 集合对象的 Close 方法关闭当前所有已打开的工作簿，具体操作代码如下。

```
Sub 关闭所有工作簿()
Workbooks.Close
End Sub
```

如果工作簿集合中含有未保存的工作簿，则会弹出是否保存更改的信息提示框，如图 8-16 所示。

【提示】不能用 Workbook 代替 Workbooks，否则在运行时会出现错误，如图 8-17 所示。



图 8-16　是否保存提示信息



图 8-17　错误提示信息

8.4.3 关闭前是否保存

利用 VBA 代码可以在程序中设置关闭前自动执行保存操作，且在执行关闭程序时不会再弹出是否保存更改的提示。常见的关闭工作簿的情况如下。

1. 关闭当前工作簿并保存更改

利用 VBA 关闭当前工作簿并保存的代码格式：

```
ActiveWorkbook().Close savechanges:=True
```

2. 关闭当前工作簿不保存更改

利用 VBA 关闭当前工作簿但不保存对其更改的代码格式：

```
ActiveWorkbook("工作簿名称").Close savechanges:=False
```

3. 关闭指定工作簿并保存更改

利用 VBA 关闭指定名称的工作簿并保存其更改的代码格式：

```
Workbooks("工作簿名称").Close savechanges:=True
```

例如，关闭名称为"新建工作簿.xls"的工作簿并保存其更改，其程序代码如下。

```
Sub 关闭并保存()
Workbooks("新建工作簿.xls").Close savechanges:=True
End Sub
```

4. 关闭指定工作簿不保存更改

利用 VBA 关闭指定名称的工作簿但是不保存对该工作簿的更改的代码格式：

```
Workbooks("工作簿名称").Close savechanges:=False
```

例如，关闭名称为“新建工作簿.xls”的工作簿并不保存对其进行的更改操作，其程序代码如下。

```
Sub 关闭并保存()
Workbooks("新建工作簿.xls").Close savechanges:=False
End Sub
```

5. 关闭所有打开的工作簿并进行保存

如果要对打开的所有工作簿进行保存并关闭操作，可执行如下代码命令。

```
Sub 全部保存并关闭()
Dim i As Workbook    '声明变量 i 为工作簿对象，该变量用于后面的循环
For Each i In Workbooks  '变量 i 遍历 Workbooks 集合对象中的所有工作簿
  If i.Name <> ThisWorkbook.Name Then    '判断每一个工作簿是否包含当前代码的工作簿
  i.Close savechanges:=True    '如果工作簿中不包含当前程序代码，则保存其更改并关闭
    End If
Next i
ThisWorkbook.Close savechanges:=True    '最后保存并关闭包含当前程序代码的工作簿
End Sub
```



8.5 综合实战 1——设置访问工作簿的属性

可以通过右击 Excel 文件，在弹出的快捷菜单中选择【属性】菜单命令的方式查看该工作簿的属性信息。打开当前工作簿的【属性】窗口后，选择【常规】选项卡即可看到该工作簿的创建时间及修改时间等属性信息，如图 8-18 所示。



图 8-18 查看文件属性

利用 VBA 程序命令同样可以查看工作簿的相关属性。比如查看工作簿上次的保存时间，具体操作命令如下，如图 8-19 所示。

```
Sub 保存时间()
Dim savetime As String
On Error Resume Next
savetime = ActiveWorkbook.BuiltinDocumentProperties("last save time").Value
    If savetime = "" Then
       MsgBox ActiveWorkbook.Name & "，该工作簿没有保存过！"
    Else
       MsgBox"您好，该工作簿最近一次的保存时间是："& savetime,,ActiveWorkbook.Name
    End If
End Sub
```

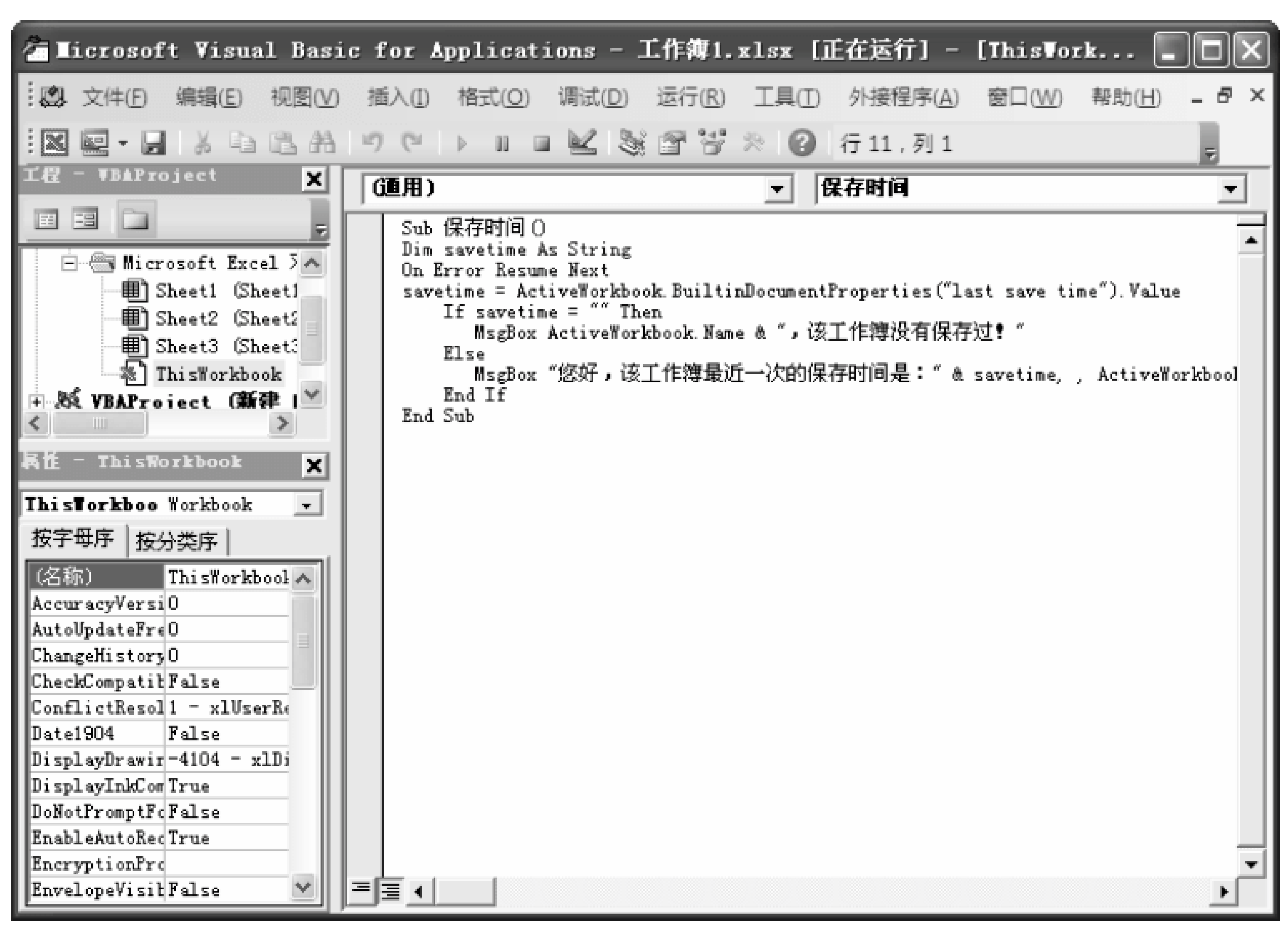


图 8-19　输入查询工作簿属性的程序

在以上代码的 If 结构语句中，系统自动对工作簿的保存时间变量“savetime”进行判断，如果“savetime”的值为空，则表示之前没有保存过这个工作簿文件，运行代码时会弹出如图 8-20 所示的提示信息。

如果工作簿之前进行过保存操作，运行以上 VBA 程序代码后，系统会自动弹出显示保存时间的提示信息，如图 8-21 所示。



图 8-20　没有保存文件的提示信息

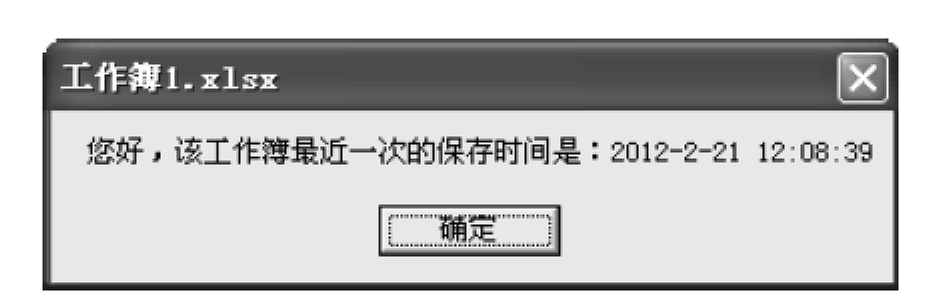


图 8-21　显示保存时间的提示



8.6 综合实战 2——禁止单元格的内容被修改

可以用锁定单元格的方式来保护工作表的内容，以防止操作者有意或无意地更改单元格内容。保护单元格内容不被更改的具体操作步骤如下。

01 在代码窗口中输入以下程序，如图 8-22 所示。

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
MsgBox "您好，该文档不能修改"
'在单元格内容发生改变时提示不能更改单元格内容
Application.EnableEvents = False
'当 Application 对象的 EnableEvents 属性为 False 时，可以禁止其他所有事件程序运行。
Target.Value = i   '使单元格清空
Application.EnableEvents = True
'由于还要使其他事件程序发作用，所以设置 Application.EnableEvents = True 让所有事件程序重新有效。
End Sub
```

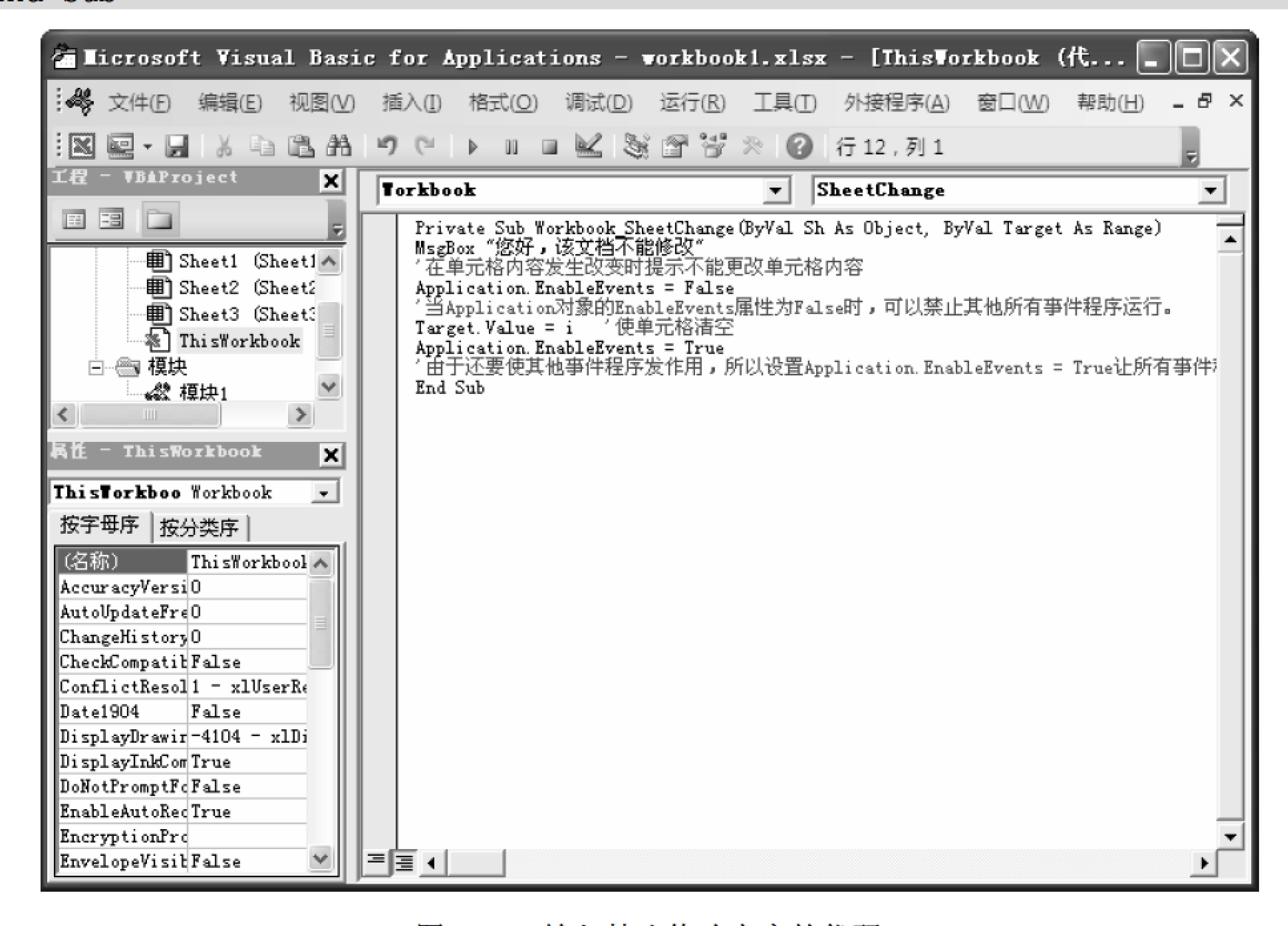


图 8-22　输入禁止修改内容的代码

02 代码输入完毕后，当更改单元格内容后，系统则弹出禁止修改单元格内容的提示，如图 8-23 所示。关闭提示信息后，单元格内容将被清空。



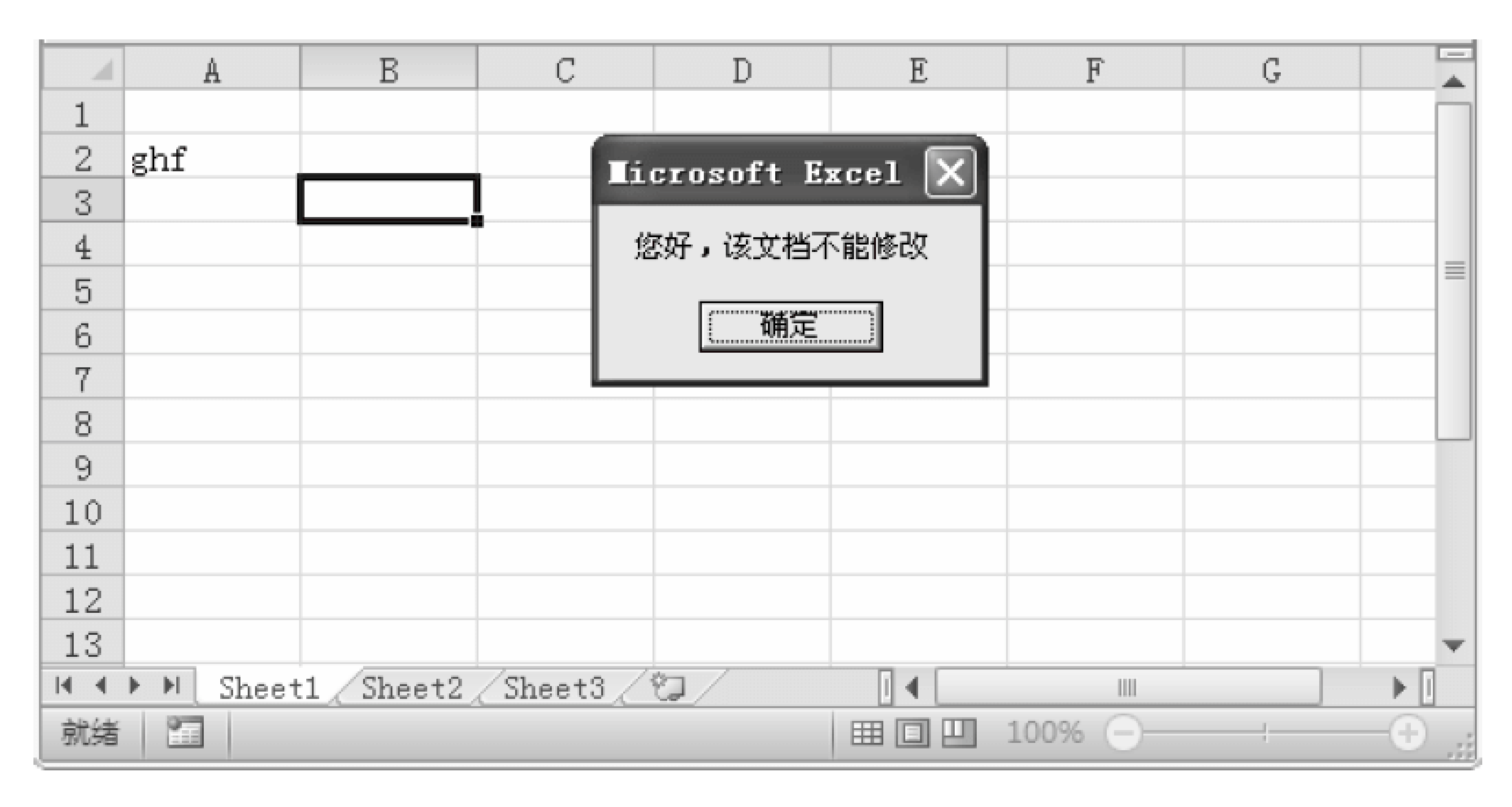


图 8-23　禁止修改内容的提示信息



8.7　综合实战 3——为工作簿添加当前日期和时间的提示信息

在 Excel VBA 中可以通过设置“工作簿打开事件”，让文件在打开时自动运行某个特定程序。

例如，在 Excel VBA 中，要实现打开 Excel 文件时，提示当前日期和具体时间，具体的操作步骤如下。

01 在代码窗口中，双击【工程】窗格中的“ThisWorkbook”选项，进入该选项对应的代码窗口，从中输入如下代码，如图 8-24 所示。

```
Private Sub Workbook_Open()
MsgBox "当前日期" & Date & ",当前时间: " & Time
End Sub
```

02 保存并关闭该文件，当再次打开该工作簿时，即可自动运行上述程序，并弹出显示当前日期和时间的提示框，如图 8-25 所示。



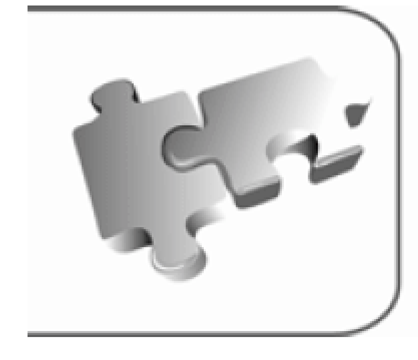


图 8-24　输入程序代码



图 8-25　打开文件时显示当前日期和时间

【提示】其中 MsgBox 函数用于弹出提示对话框，Date 函数用于返回系统的当前日期，Time 函数用于返回系统的当前时间。



8.8　综合实战 4——禁止打印工作簿文件内容

利用 VBA 代码可以实现禁止相关工作簿的打印操作。其中工作簿事件列表中的 Workbook_BeforePrint 事件和打印动作有关联，可对其进行设置禁用打印工作簿。禁止打印工作簿文件内容的具体操作步骤如下。

01 在 VBA 代码窗口中，双击【工程】窗口中的"ThisWorkbook"选项，进入到该选项对应的代码窗口中，在"对象"下拉列表中选择"Workbook"选项，在"过程"下拉列表中选择"BeforePrint"选项，然后在其代码框架中输入相应的程序代码，最终代码如下，如图 8-26 所示。

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
    Cancel = True  'Cancel 作为事件提供的自变量，当值设置为 True 时，打印功能失效
    MsgBox "您好，打印功能已被禁止！", vbExcelamation
'用 MsgBox 函数提供提示框，告诉用户不能打印文档内容
End Sub
```

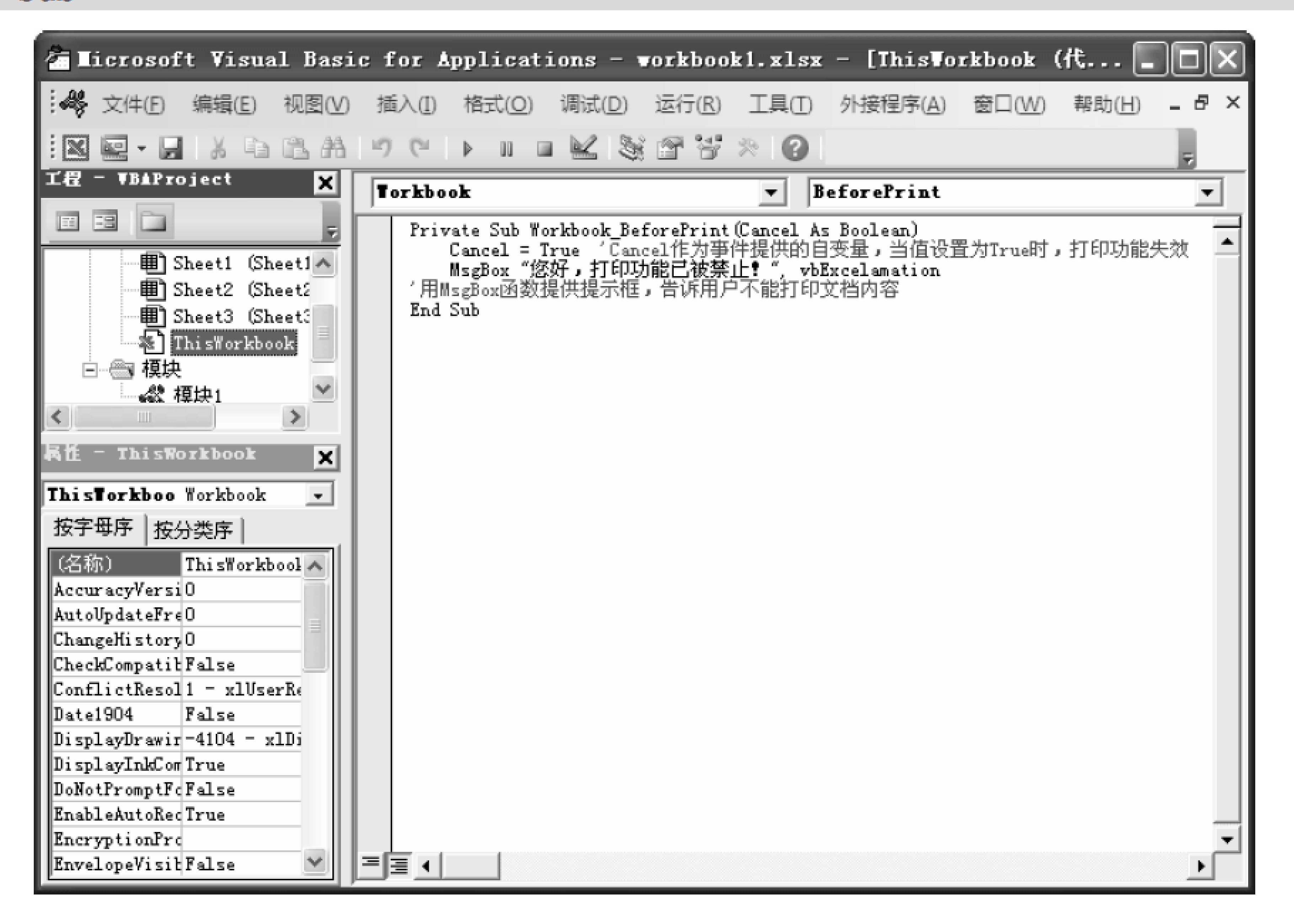


图 8-26　输入禁止打印功能的代码

02 代码输入完毕后，当在工作表中对该文件进行打印时，系统则会弹出相应的提示信息，如图 8-27 所示。



图 8-27　禁止打印提示信息






8.9 高手私房菜

技巧 1：禁止使用关闭按钮

通过设置工作簿的 BeforeClose 事件，可以让 Excel 窗口右上角的【关闭】功能失效，具体代码如下。

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
Cancel = True
End Sub
```

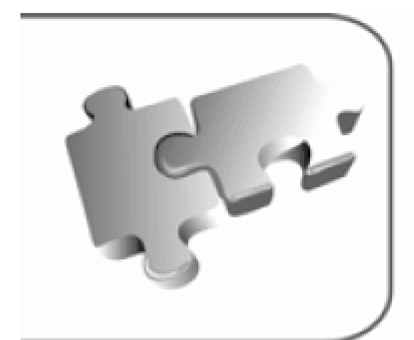
其中，Workbook_BeforeClose 是当工作簿关闭时发生的事件；Cancel As Boolean 是工作簿关闭事件提供的自变量，如果 Cancel = True，则【关闭】按钮失效。

技巧 2：轻松解决因为密码设置的错误提示

使用 Excel VBA 代码执行打开已设置了密码的工作簿时，需要注意路径和文件名是否正确，如果这些都正确了，还应注意代码是否有书写错误，其中需要特别注意的是在 password 参数后并不是直接使用“=”号连接密码的，而是使用“:=”。

第 9 章　Excel 工作表相关操作

工作表是操作 Excel 时必不可少的对象，同时是编辑单元格对象的入口和载体。要想利用 Excel VBA 更好地控制 Excel 工作簿和单元格，掌握控制工作表的方法至关重要。本章将详细介绍 Excel 工作表的相关操作，如工作表的插入、选择、隐藏及删除等。



9.1 选择工作表

9.1.1 选取指定名称的工作表

在利用 VBA 进行程序设计时，有些操作需要在指定的工作表中完成，所以应该先选择指定的工作表。

【例 9.1】在工作簿中选择名称为“2 月份工资表”的工作表，具体操作命令如下。

```
Sub 选择指定工作表()
  Worksheets("2 月份工资表").Select
End Sub
```

代码分析：其中 Select 是选择工作表集合对象（Worksheets）或工作表对象（Worksheet）的方法，其功能是使工作表处于选中状态。

9.1.2 选择多个工作表

在工作表操作中，经常会遇到同时选择多个工作表的操作，这可以使用 Shift 键或 Ctrl 键与鼠标配合的方式来实现，使用 Shift 键可以选择连续的几张表，而按 Ctrl 键同时单击可选择多个不连续的表。本节主要介绍使用 VBA 命令的方式实现多个工作表的选择。

【例 9.2】同时选中两个不连续的工作表“Sheet1”和“Sheet3”，具体操作命令如下。

```
Sub 选择多个工作表()
  Worksheets(Array("Sheet1", "Sheet3")).Select
End Sub
```

代码分析：其中 Array 函数用于指定要选择的多个工作表，工作表的名称必须在英文输

入状态下的双引号内，且工作表之间用逗号分隔。

【例 9.3】选择工作簿中的第 2 个到第 5 个工作表，具体操作命令如下。

```
Sub 选择多个工作表()
   Worksheets(Array(2,5)).Select
End Sub
```

运行上述代码可以选中工作簿中的 4 个工作表。

【例 9.4】选中工作簿中的所有工作表，具体操作命令如下。

```
Sub 选择全部工作表()
   Worksheets.Select
End Sub
```

【提示】上述程序中的“Worksheets.Select”语句可以用“Sheets.Select”替换，效果是一样的。



9.2 插入工作表

一般新建的工作簿中默认带有 3 个工作表，分别是“Sheet1”、“Sheet2”和“Sheet3”，当工作表的数量不够用时，可以插入工作表。

9.2.1 插入一个工作表

利用 VBA 插入工作表，主要是使用了 Worksheets 集合对象的 Add 命令。可以使用以下程序代码在当前活动工作表的前面插入单个空白工作表。

```
Sub 插入单个工作表()
   Worksheet.Add
End Sub
```

如果要插入指定名称的单个工作表，可以设定一个名称变量，并对该变量赋予需要的工作表名称。

【例 9.5】在工作簿中插入名为“考勤表”的工作表，具体操作命令如下，如图 9-1 所示。

```
Sub 插入考勤表()
   Dim i As Object    '声明 i 为对象变量
   Set i = Worksheets.Add   '将新添加的工作表对象赋予对象变量 i
   i.Name = "考勤表"   'Name 是工作表对象的属性，用于指定工作表的名称
End Sub
```

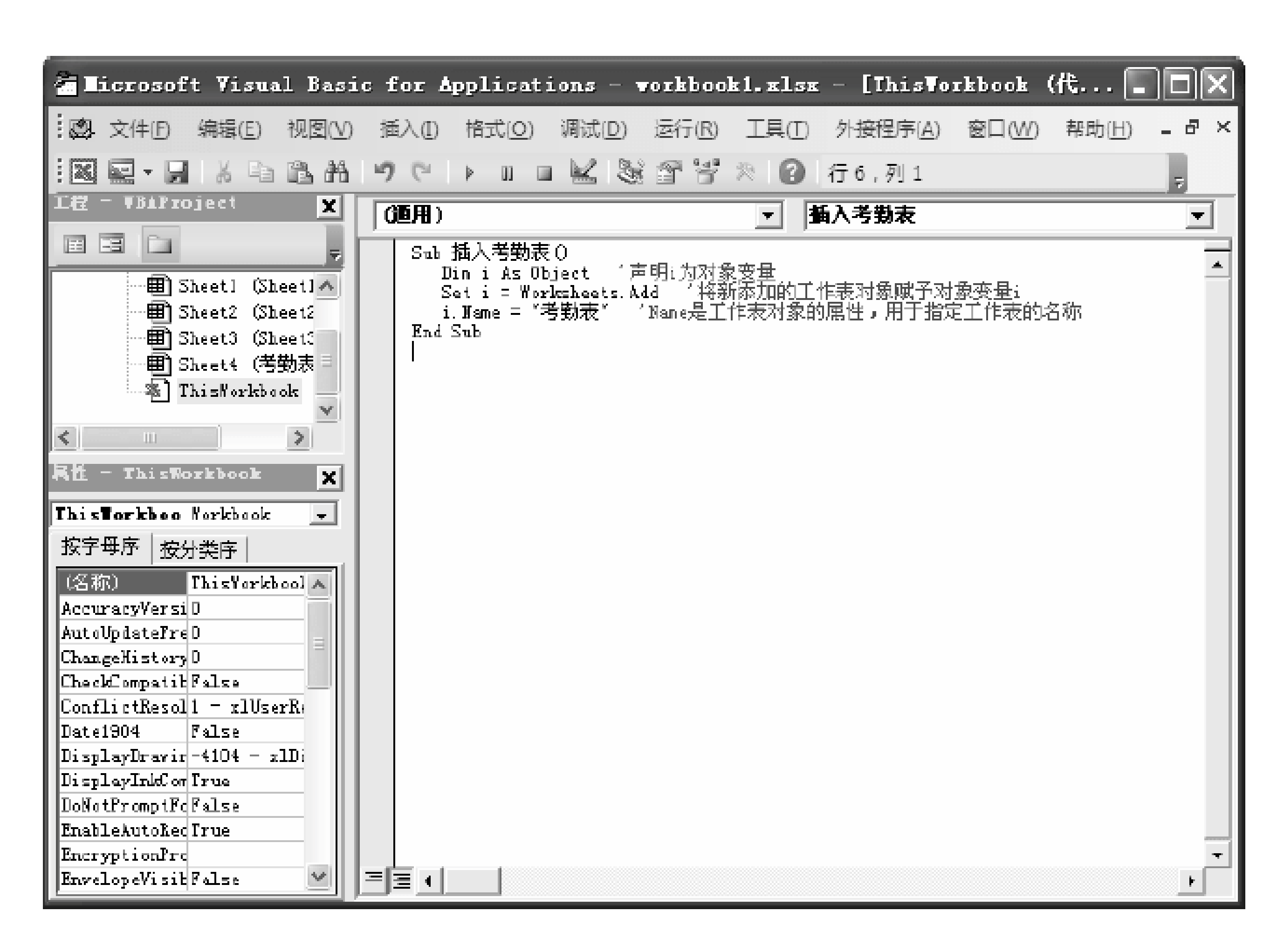


图 9-1　输入插入指定名称的单个工作表程序

运行上述程序，即可在当前活动工作表的前面插入一个名为“考勤表”的工作表，如图 9-2 所示。

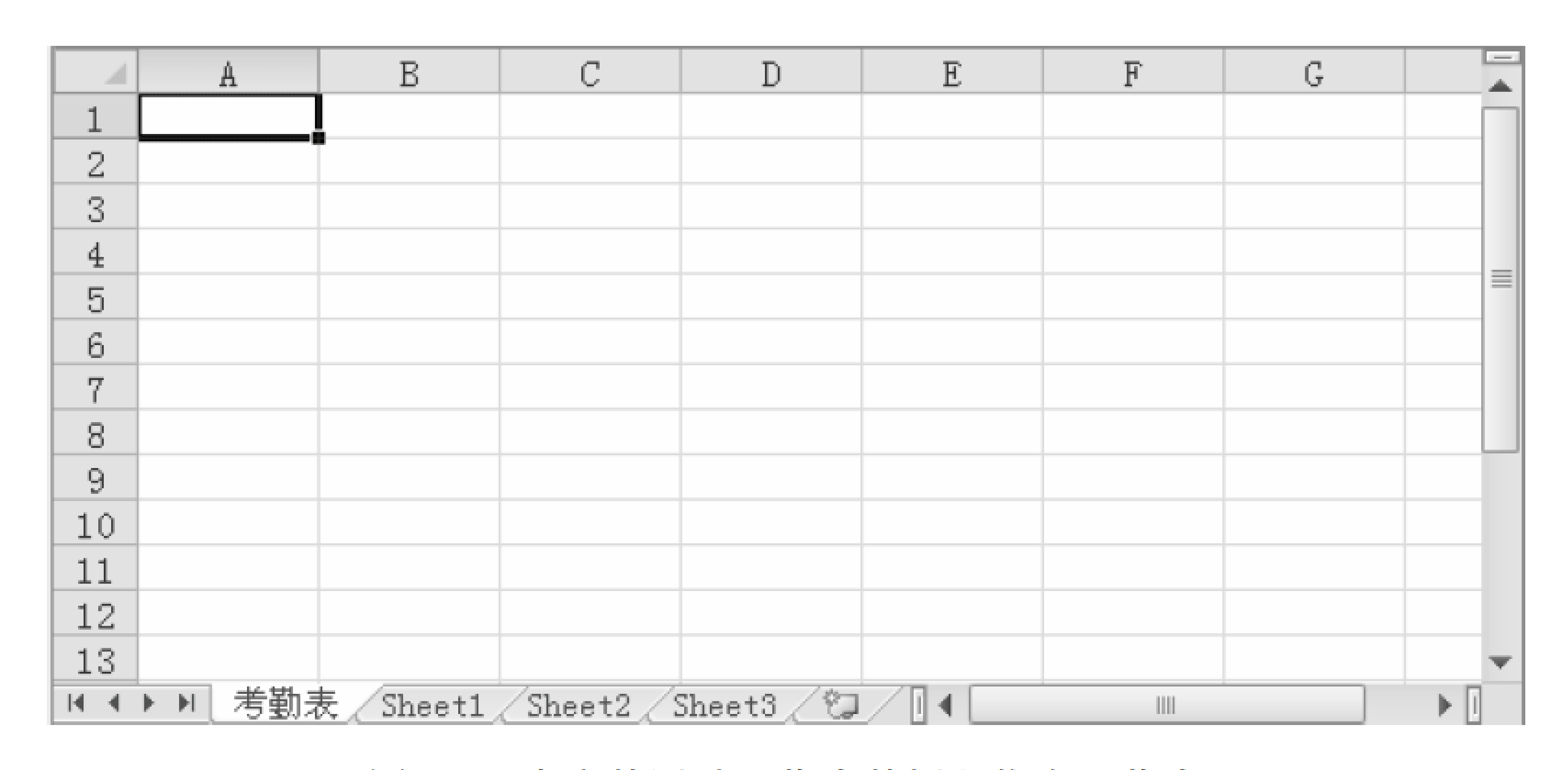


图 9-2　在当前活动工作表前插入指定工作表

除此之外，还可以使用插入新工作表并为当前活动工作表命名的方式插入指定名称的工作表，具体操作命令如下。

```
Sub 插入单个工作表并命名()
   Sheets.Add
   ActiveSheet.Name = "工资表"    '当前活动工作表的名称为“工资表”
End Sub
```

代码分析：ActiveSheet 是指当前活动工作表。因为新插入的工作表会自动成为活动工作表，所以 ActiveSheet 是指新插入的工作表。

9.2.2　插入多个工作表

在使用 Excel 时，经常遇到需要同时插入多张表的情况，如果逐个插入会比较麻烦，使

用 VBA 中 Worksheets 集合对象的 Count 属性可以实现同时插入多张空白工作表。

【例 9.6】使用 VBA 命令同时插入 4 个工作表，具体操作命令如下。

```
Sub 插入多个工作表()
   Worksheets.Add Count:=4
'插入工作表的数量为 4，这里使用了自变量 Count 来控制插入工作表的数量
End Sub
```

如果在插入多个工作表时想同时为这些工作表指定名称，可以使用变量循环的方式为工作表指定具有一定顺序的工作表名。

【例 9.7】插入多个工作表，并分别将其命名为“2011 年”～“2015 年”，具体操作命令如下，如图 9-3 所示。

```
Sub 插入多个工作表并命名()
   Dim i As Integer  '声明 i 为整数变量
   For i = 1 To 5  '设置循环次数是 5 次，插入（Sheets.Add）的工作表数量也为 5 个
      Sheets.Add
      ActiveSheet.Name = "201" & i & "年"
   Next i
End Sub
```

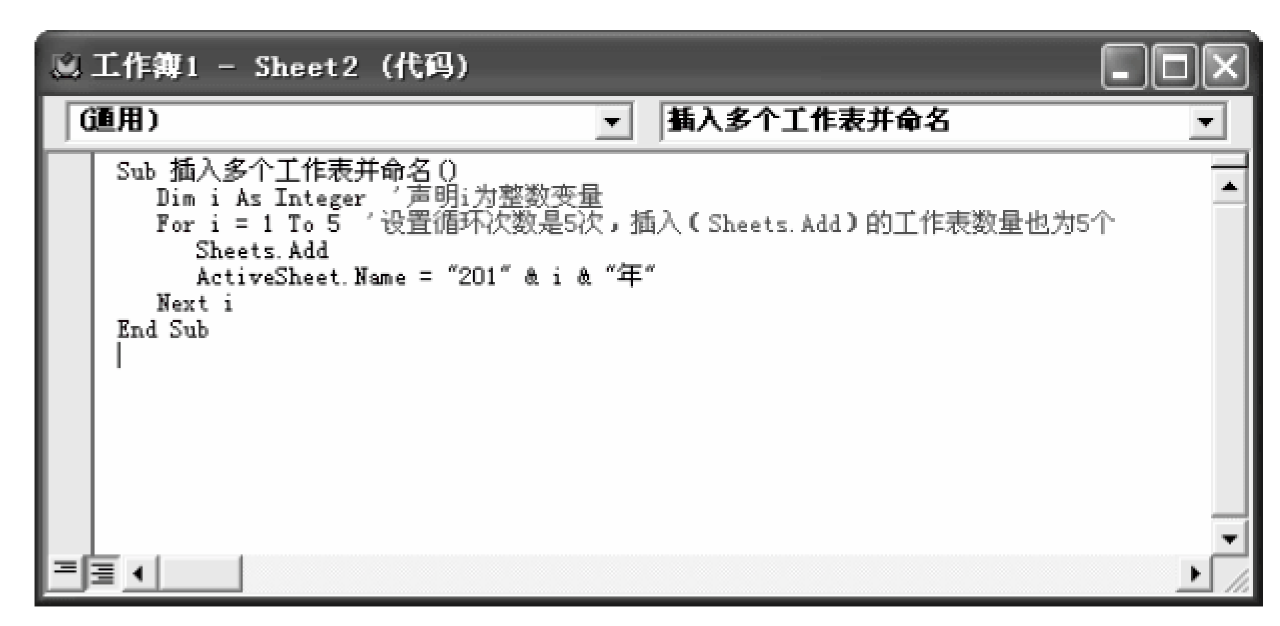


图 9-3　实现插入多个工作表并命名的程序代码

运行上述程序，即可在当前活动工作表前面插入 5 个工作表，分别命名为“2011 年”～“2015 年”，如图 9-4 所示。

图 9-4　插入多个工作表并命名

9.2.3 在指定位置插入工作表

可以根据需要在指定位置插入工作表。根据插入工作表的位置不同，可以分为以下 4 种情况。

1. 在指定工作表前插入工作表

【例 9.8】使用自变量 Before 在已有工作表 Sheet1 前插入一个新工作表，具体操作命令如下。

```
Sub 指定位置前插入()
   Worksheets.Add Before:=Sheets("Sheet1")
End Sub
```

运行该程序可以实现在工作表 Sheet1 前插入一张空白工作表。

2. 在指定工作表后插入工作表

【例 9.9】使用自变量 After 在已有工作表 Sheet1 后插入一个新工作表，具体操作命令如下。

```
Sub 指定位置后插入()
   Worksheets.Add After:=Sheets("Sheet1")
End Sub
```

运行该程序可以实现在工作表 Sheet1 后插入一张空白工作表。

3. 在工作表最前位置插入工作表

【例 9.10】在所有工作表最前位置插入工作表，具体操作命令如下。

```
Sub 最前位置插入()
   Worksheets.Add Before:=Sheets(1)
End Sub
```

4. 在工作表最后位置插入工作表

【例 9.11】在已有工作表最后位置插入新工作表，具体操作命令如下。

```
Sub 最后位置插入()
   Worksheets.Add After:=Sheets("Sheets.Count")
End Sub
```

这里利用“Sheets.Count”来确定当前工作簿中工作表的数量。

9.2.4 插入前判断工作表是否存在

插入工作表时可以为表定义名称，如果插入的表名已经存在，很可能造成命令错误。所以在插入指定名称的工作表前需要判断该工作表是否存在，然后再进行插入工作表的操作。

【例 9.12】在工作簿中插入“2 月份工资表”工作表，判断表“2 月份工资表”是否存在，并执行插入操作的命令如下。



在 VBA 代码窗口中，新建模块 1 并从中输入如下程序代码，如图 9-5 所示。

```
Sub 插入前判断()
Dim x As Integer            '设置循环变量
For x = 1 To Sheets.Count  '设置变量在“1”到“工作表数量”之间循环
  If Sheets(x).Name = "2 月份工资表" Then    '判断工作表的名称是否为“2 月份工资表”
    MsgBox "已存在 2 月份工资表"     '当“2 月份工资表”工作表存在时，运行该程序会弹出显示“已存在 2 月份工资表”
    Exit Sub        '如果存在“2 月份工资表”工作表，则退出该过程
  End If
Next x
    Sheets.Add       '如果不存在则插入工作表
    ActiveSheet.Name = "2 月份工资表"   'ActiveSheet 是指当前活动工作表，所以对当前活动工作表命名即可
End Sub
```

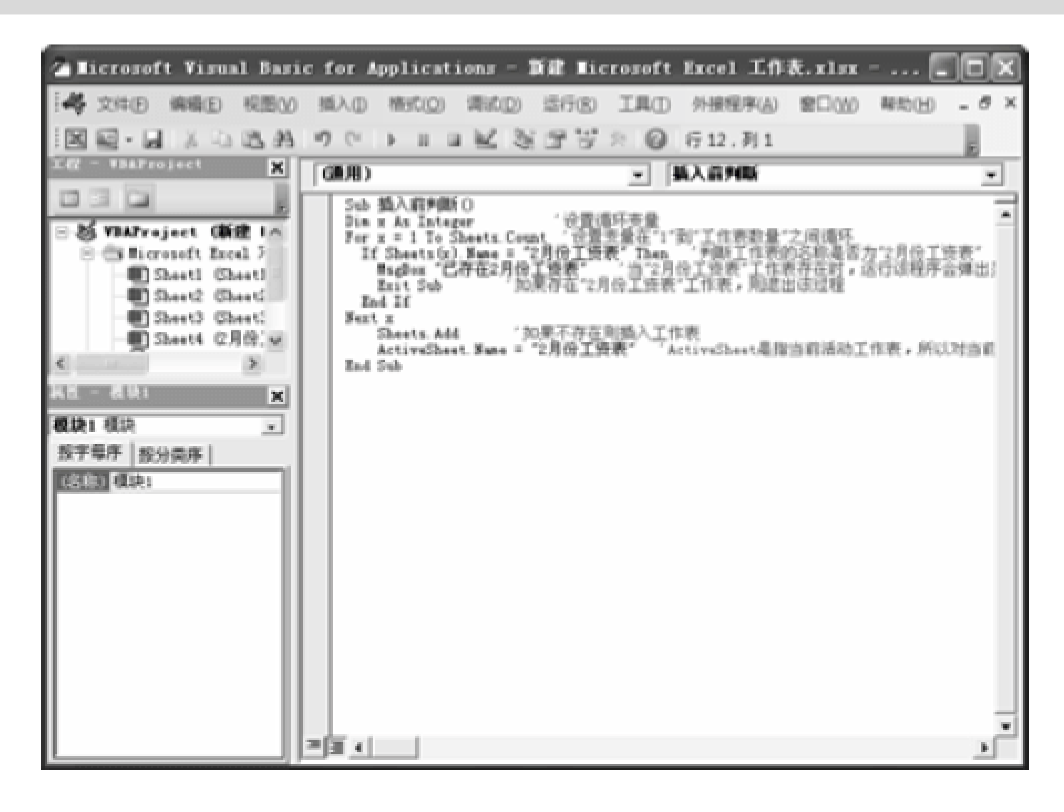


图 9-5　输入代码

程序输入完毕后，运行该程序，若“2 月份工资表”工作表存在，运行该程序会弹出显示“已存在 2 月份工资表”信息提示框，如图 9-6 所示。

在插入指定名称的工作表时，如果不进行判断，取消上述代码中的 For 循环，则在插入名称相同的工作表时会弹出如图 9-7 所示的错误提示信息。

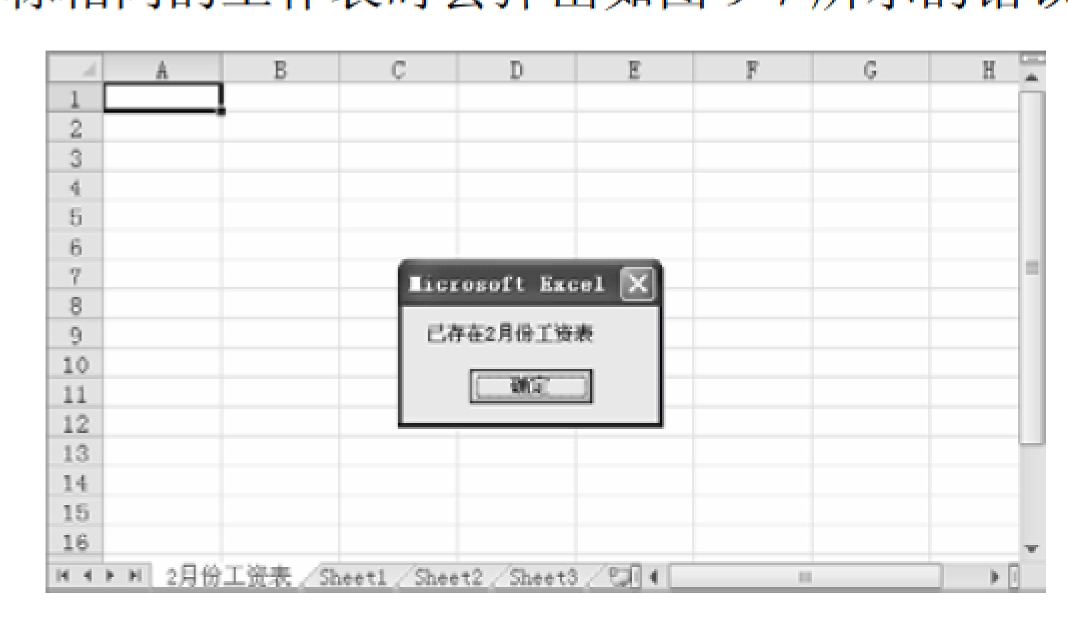
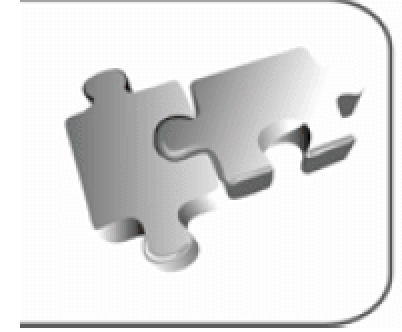


图 9-6　已存在提示信息



图 9-7　工作表重复命名的错误提示



9.3 移动工作表

用户可以通过移动的方式改变工作表在工作簿中的位置，或者将工作表移动到其他工作簿中。

9.3.1 在工作簿内移动工作表

在工作簿内移动工作表有两种情况，分别是移动到指定工作表之前和移动到指定工作表之后。

1．移动到指定工作表之前

【例 9.13】将工作表“考勤表”移动到工作表“工资表”之前，具体操作命令如下。

```
Sub 移动到工作表之前()
   Sheets("考勤表").Move before:=Sheets("工资表")
End Sub
```

代码分析：其中 Move 起到移动工作表对象的作用，before 是设置移动到指定工作表之前。

2．移动到指定工作表之后

【例 9.14】将工作表“考勤表”移动到工作表“工资表”之后，具体操作命令如下。

```
Sub 移动到工作表之后()
   Sheets("考勤表").Move after:=Sheets("工资表")
End Sub
```

其中的 after 表示移动到指定工作表之后。

9.3.2 工作簿之间移动工作表

工作簿之间移动工作表分为移动到指定工作簿和移动到新工作簿。

1．移动到指定工作簿

【例 9.15】将工作表“4 月份考勤表”移动到“第一季度考勤.xls”工作簿的第 3 个工作表的后面，具体操作命令如下。

```
Sub 移动到指定工作簿()
   Sheets("4 月份考勤表").Move after:=Workbooks("第一季度考勤表.xls").Sheets(3)
End Sub
```



【提示】执行上述命令时必须保证移动到的目标工作簿处于打开状态，否则运行程序时会弹出错误提示信息，如图 9-8 所示。



图 9-8　错误提示信息

2. 移动到新工作簿

【例 9.16】将工作表“考勤表”移动到一个新的工作簿中，具体操作命令如下。

```
Sub 移动到新工作簿()
   Worksheets("考勤表").Move
End Sub
```

当 Move 方法后没有自变量时，即将工作表移动到一个新建的工作簿中。



9.4 复制工作表

当需要添加相同的工作表时，不需要重复新建，只需要复制操作即可快速达到相同的效果。

9.4.1 在工作簿内复制工作表

在工作簿内复制工作表有两种情况，分别是复制到指定工作表之前和复制到指定工作表之后。

1. 复制到指定工作表之前

【例 9.17】将“工资表”复制到“考勤表”之前，具体操作命令如下。

```
Sub 复制到指定工作表之前()
   Sheets("工资表").Copy before:=Sheets("考勤表")
End Sub
```

【提示】执行复制时，产生的新表表名为“工资表（2)”，如果再执行一次该程序，产生的新表表名为“工资表（3)”，以此类推。

2. 复制到指定工作表之后

【例 9.18】将“工资表”复制到“考勤表”之后，具体操作命令如下。

```
Sub 复制到指定工作表之后()
   Sheets("工资表").Copy after:=Sheets("考勤表")
End Sub
```

运行上述程序可以在“考勤表”后面增加“工资表”的副本，如图 9-9 所示。

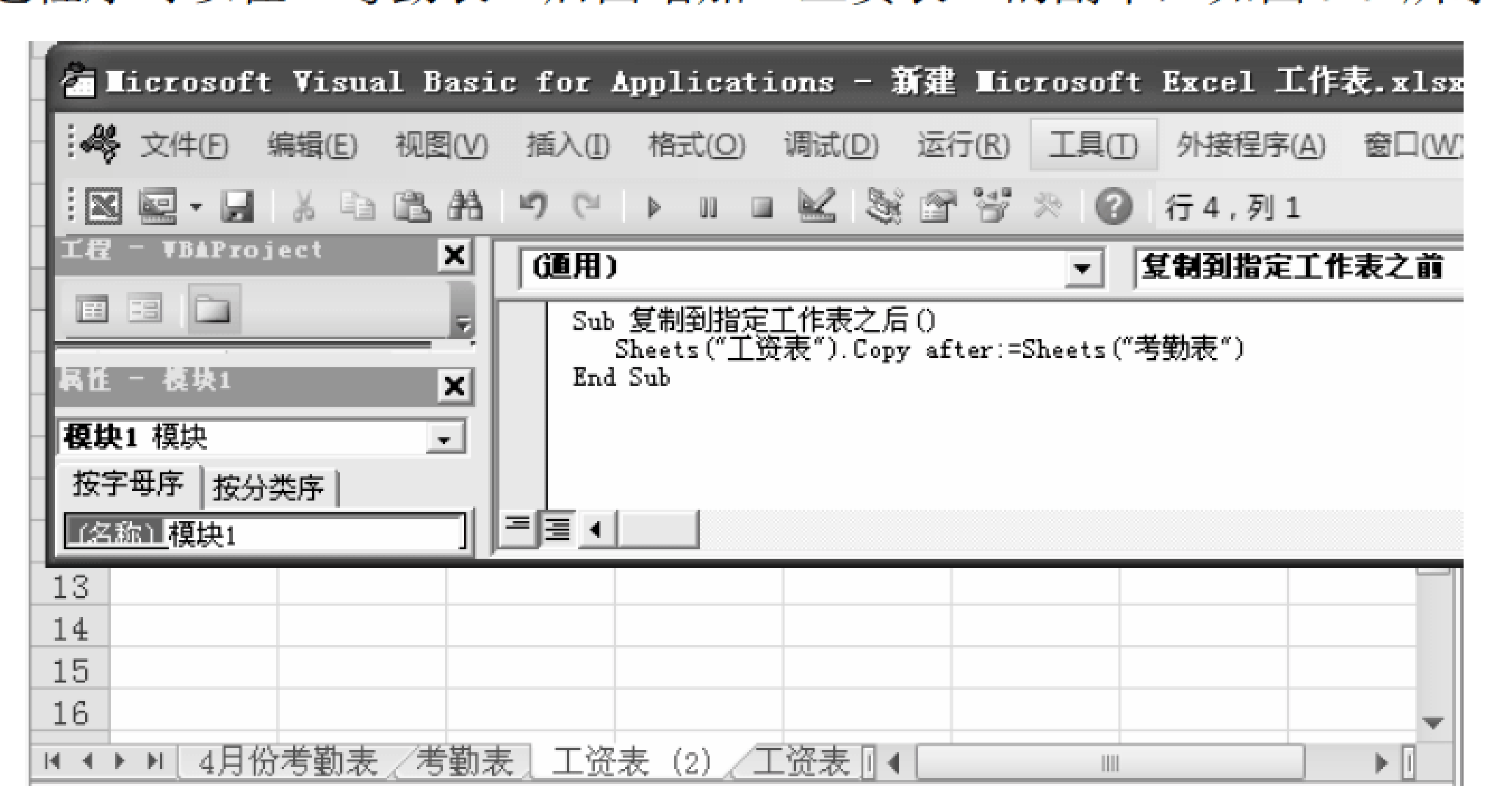


图 9-9　复制到指定工作表之后

9.4.2　在工作簿间复制工作表

在工作簿间复制工作表分为两种情况，分别是复制到指定工作簿和复制到新建工作簿。

1. 复制到指定工作簿

【例 9.19】将工作表“4 月份考勤表”复制到“第一季度考勤.xlsx”工作簿中的第 3 个工作表后，具体操作命令如下。

```
Sub 复制到指定工作簿()
   Sheets("4 月份考勤表").Copy after:=Workbooks("第一季度考勤.xlsx").Sheets(3)
End Sub
```

2. 复制到新建工作簿

【例 9.20】将工作表“考勤表”复制到一个新的工作簿中，具体操作命令如下。

```
Sub 复制到新的工作簿()
   Sheets("考勤表").Copy
End Sub
```

【提示】复制的目标工作簿必须处于打开状态。

9.4.3 将工作表另存为工作簿

如果要把一个工作簿中的所有工作表分别另存为一个工作簿文件，需要用到工作表对象的“复制”和工作簿的“另存为”属性。

【例 9.21】将工作簿中的 4 个工作表分别保存为 4 个独立的工作簿文件，存放到 D 盘，生成的工作簿名和原来的工作表名相同，具体操作命令如下。

```
    Sub 工作表另存工作簿()
    Dim x As Workbook    '设置 x 为工作簿对象变量
    Set x = ActiveWorkbook    '把当前工作簿赋予变量 x
      Dim y As Integer    '声明 y 为整型变量，用于表示工作簿中工作表的索引号
       For y = 1 To x.Worksheets.Count    '设置循环范围为当前活动工作簿中的所有工作表
        x.Worksheets(y).Copy    '复制工作表
        ActiveWorkbook.SaveAs Filename:="D:\ excel 2010\" & x.Worksheets(y).Name
& ".xls"
    '将新建的工作簿另存到 D 盘，并且名称与工作表名称相同
       ActiveWindow.Close    '关闭当前窗口，即新建的工作簿
       Next y
    End Sub
```

运行上述程序后，即可将 4 个工作表分别保存在 D:\excel 文件夹中，如图 9-10 所示。



图 9-10 将工作表另存为工作簿



9.5 删除工作表

在工作表不需要的情况下，可以将其删除。本节将详细介绍工作表删除的操作方法和技巧。

9.5.1　删除指定的工作表

在 Excel 中删除工作表的方法通常是在工作表标签上单击鼠标右键，在弹出的快捷菜单中选择【删除】菜单命令。同时也可以使用 VBA 中工作表对象的 Delete 方法，实现删除工作表的操作。

【例 9.22】删除工作表“工资表”，具体操作命令如下。

```
Sub 删除指定工作表()
   Sheets("工资表").Delete
End Sub
```

运行上述程序后，系统会弹出提示用户确认删除的对话框，如图 9-11 所示。单击【删除】按钮，即可删除指定工作表。

【提示】删除后要保证工作簿中至少保留一个工作表，否则会弹出删除工作表时的错误提示信息，如图 9-12 所示。

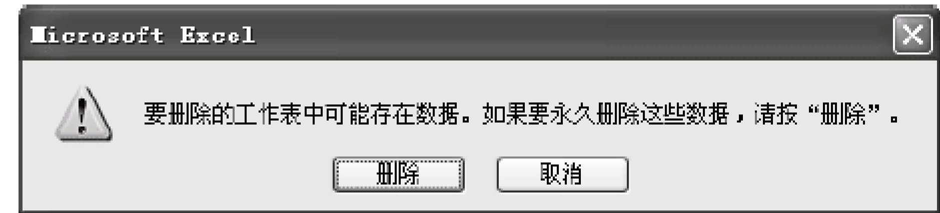


图 9-11　确认删除提示信息

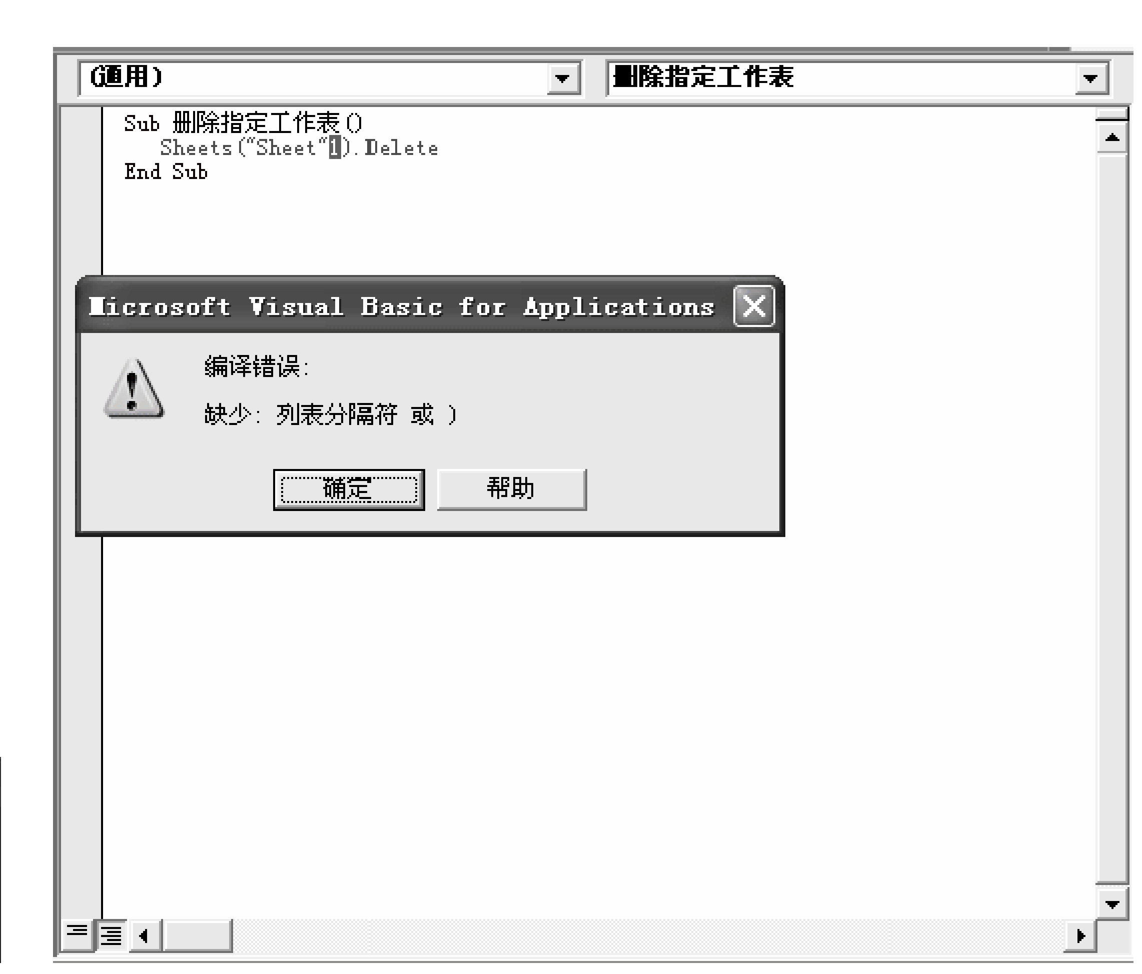


图 9-12　删除工作表时的错误提示

在执行工作表删除时，可以使用 VBA 程序检测是否删除成功，这主要是通过单击【删除】或【取消】按钮时返回的 Boolean 类型值决定的。

【例 9.23】删除“工资表”，并执行删除检测，返回提示信息，具体操作命令如下。

```
Sub 检测是否成功删除指定工作表()
Dim x As Boolean
   x = Worksheets("工资表").Delete
   If x = True Then
      MsgBox "删除表格操作成功完成！", vbInformation
   Else
     MsgBox "删除表格操作被取消！", vbInformation
   End If
End Sub
```



命令执行后，弹出删除提示框，单击【取消】按钮，弹出取消删除提示信息，如图 9-13 所示；单击【删除】按钮，弹出删除表格成功提示框，如图 9-14 所示。




图 9-13 取消删除提示信息



图 9-14 成功删除提示信息

另外，在删除工作表时，用户完全可以取消弹出提示用户确认删除的对话框，具体操作命令如下。

```
Sub 不显示确认删除的对话框()
   Application.DisplayAlerts = False  '禁止应用程序显示警告和消息功能
   For x = 1 To Worksheets.Count
      Worksheets("工资表").Delete
   Application.DisplayAlerts = True   '恢复应用程序显示警告和消息功能
   Next x
End Sub
```

Application 对象的 DisplayAlerts 属性用于指示在宏运行时 Excel 是否显示特定的警告或消息对话框，其默认值为 True。

9.5.2 删除未选中的工作表

如果要保留当前活动的所有工作表，而删除其他未选中的工作表，具体操作命令如下。

```
Sub 删除未选中的工作表()
Dim x As Integer
Dim y As Integer
    x = ActiveWindow.SelectedSheets.Count   '确定当前活动窗口中选中工作表的数量
    A ctiveWindow.SelectedSheets.Move before:=Sheets(1)
'将当前选中的所有工作表移动到最前面
    Application.DisplayAlerts = False
    For y = Sheets.Count To x + 1 Step -1
'设置循环范围为从后到前，即从最后一个工作表到第 x+1 个工作表之间的所有工作表
        Worksheets(y).Delete
    Next y
    Application.DisplayAlerts = True
End Sub
```

这段程序的功能是先将窗口中选中的工作表移动到最前面，然后利用循环语句删除其后面的工作表，即实现了删除未选中的工作表的功能。



9.6 隐藏和取消隐藏工作表

在工作表不使用的情况下，可以将其隐藏。本节将详细介绍工作表隐藏的操作方法和技巧。

9.6.1 隐藏工作表

隐藏工作表包括隐藏一个工作表和隐藏多个工作表，用户也可以按照指定的条件进行隐藏。

1. 隐藏一个工作表

【例 9.24】利用 VBA 将已有工作表“工资表”隐藏，具体操作命令如下。

```
Sub 隐藏一个工作表()
   Sheets("工资表").Visible = 0  '指工作表的显示属性
End Sub
```

Visible 是工作表对象的属性，当其值为“0”或“xlSheetHidden”时，会将工作表隐藏，但此时可以在工作簿中选择【开始】➢【格式】➢【取消隐藏工作表】菜单命令来取消该工作表的隐藏。如果将 Visible 的值设为“2”或“xlSheetVeryHidden”，则会将工作表彻底隐藏，并且在工作簿中通过选择【开始】➢【格式】➢【取消隐藏工作表】菜单命令，也无法取消该工作表的隐藏。

2. 隐藏多个工作表

【例 9.25】将工作表“工资表”和“考勤表”隐藏，具体操作命令如下。

```
Sub 隐藏多个工作表()
   Sheets(Array("工资表", "考勤表")).Visible = 0
End Sub
```

3. 按指定条件隐藏

利用循环语句可以判断出满足特定条件的工作表，并对其实施隐藏操作。

【例 9.26】将工作簿中“工资表”之外的所有工作表隐藏，具体的操作命令如下。

```
Sub 指定条件隐藏()
Dim x As Integer
For x = 1 To Sheets.Count    '设置变量 x 在所有工作表之间循环
  If Sheets(x).Name <> "工资表" Then    '判断工作表的名称是否为“工资表”，若不是则执行'下一条语句
    Sheets(x).Visible = 0       '符合条件判断条件的工作表隐藏
  End If
Next x
End Sub
```

9.6.2 取消工作表的隐藏

取消工作表的隐藏也是通过设置 Visible 属性值来实现的。

【例 9.27】取消单个工作表“工资表”的隐藏，具体操作命令如下。

```
Sub 取消单个工作表的隐藏()
    Sheets("工资表").Visible = -1
End Sub
```

【例 9.28】取消工作表“工资表”和“考勤表”的隐藏，具体操作命令如下。

```
Sub 取消多个工作表的隐藏()
    Sheets(Array("工资表", "考勤表")).Visible = -1
End Sub
```

【例 9.29】取消工作簿中所有工作表的隐藏，具体操作命令如下。

```
Sub 取消所有工作表的隐藏()
    Dim x As Integer
    For x = 1 To Sheets.Count
        Sheets(x).Visible = -1
    Next x
End Sub
```

这里用一个循环来设置每一个工作表的显示，即属性值为-1，可以逐个取消工作表的隐藏。如果将 Visible 的属性值设置为“xlSheetVisible”，也可以将工作表显示出来。



9.7 提供工作表的安全性

在 Excel 基本操作中，用户可以对工作表实施保护，以避免工作表中的已有数据被修改。

9.7.1 保护单个或所有工作表

保护工作表可以使用功能区【审阅】选项卡中【更改】组的【保护工作表】工具按钮，在打开的【保护工作表】对话框中进行保护工作表的操作，如图 9-15 所示。

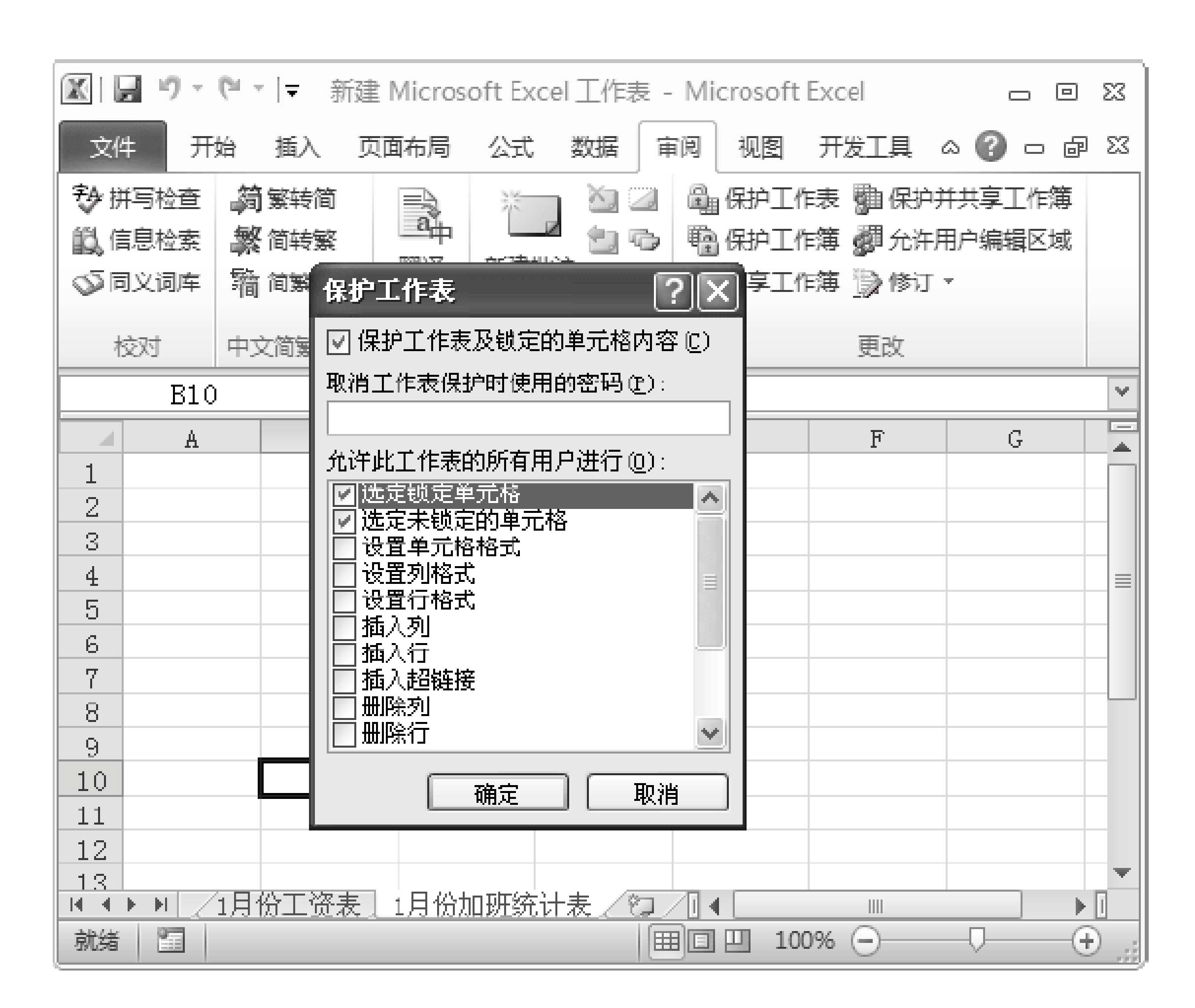


图 9-15　【保护工作表】对话框

利用 VBA 代码设置保护工作表的操作也很简单，主要分为以下两种。

1. 保护单个工作表

【例 9.30】保护单个工作表“工资表”，具体操作命令如下。

```
Sub 保护单个工作表()
   Worksheets("工资表").Protect Password:=abc123
End Sub
```

代码分析：其中使用了工作表对象的 Protect 方法设置工作表的保护，并将 Password 密码属性值设置为“abc123”。

【提示】 在设置保护密码时，如果使用的是字母，应注意区分字母的大小写。

2. 保护所有工作表

【例 9.31】同时为工作簿中的所有工作表设置保护，并指定保护密码，具体操作命令如下。

```
Sub 保护所有工作表()
Dim x As Integer
   Worksheets(x).Protect Password:=abc123
  Next x
End Sub
```

9.7.2 解除工作表的保护

利用 VBA 也可以很轻松地解除工作表的保护。

【例 9.32】解除工作表“工资表”的保护，具体操作命令如下。

```
Sub 工作表解除保护()
   Sheets("工资表").Unprotect Password:=abc123
End Sub
```

与 Protect 方法相对应，工作表的 Unprotect 方法是解除工作表的保护。

如果将工作簿中的所有工作表设置了同样的保护密码，则用户可以利用 VBA 一次解除所有工作表的保护，具体操作命令如下。

```
Sub 解除所有工作表保护()
Dim i As Integer
  For i = 1 To Sheets.Count
    Sheets(i).Unprotect Password:=abc123
  Next i
End Sub
```



9.8 综合实战 1——限制工作表的滚动区域

通过设置 Worksheet 对象的 ScrollArea 属性可以限制工作表允许滚动的区域。当设置了工作表滚动区域后，用户不能选定滚动区域之外的单元格，同时，工作表的一些相应功能也可能被禁止，例如选中整行或整列等。

可以通过【属性】窗口和 VBA 代码来实现 Worksheet 对象的 ScrollArea 属性的设置，具体内容介绍如下。

1. 在【属性】窗口中设置

通过修改 VBA 界面工作表的【属性】信息，可以限制工作表的滚动区域，具体操作步骤如下。

01 在工作簿中，首先确定允许滚动的区域，这里将工作表 Sheet1 中的 "B3:K10" 单元格区域设置为允许滚动的区域，如图 9-16 所示。

02 按 Alt+F11 组合键，打开 VBA 代码窗口，在【工程资源管理器】窗口中选中 "Sheet1" 选项，然后在其【属性】窗口中切换到【按字母序】选项卡，在 "ScrollArea" 文本框中输入 "$B$3:$K$10"，如图 9-17 所示。

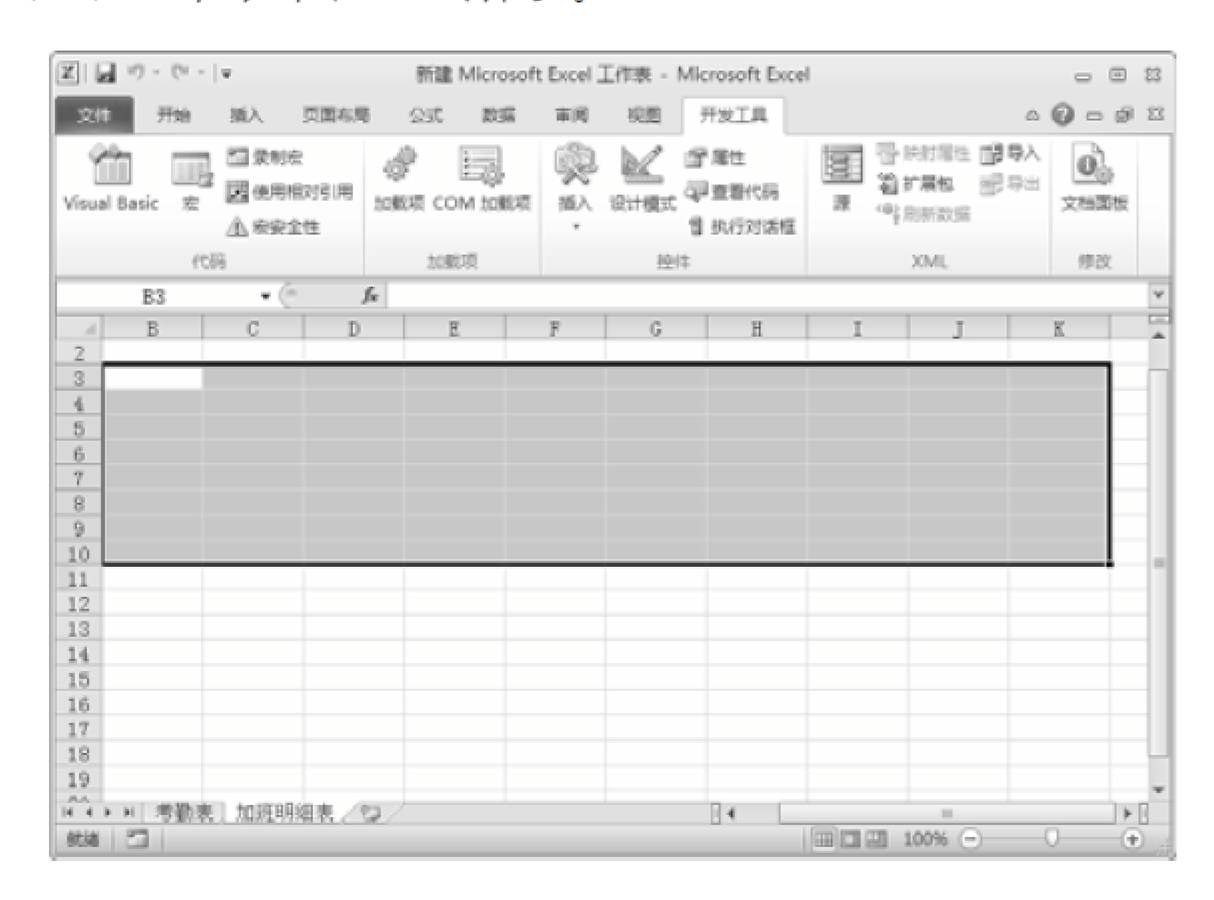


图 9-16　设置允许滚动的区域

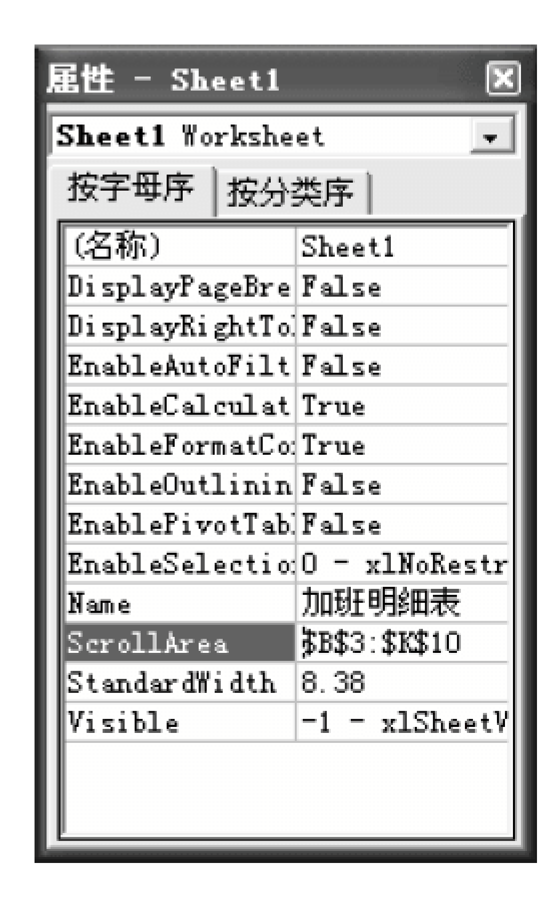


图 9-17　设置【属性】窗口

03 返回工作表，此时即可将〝C5：H14〞单元格区域设置为允许滚动的区域，而区域之外的单元格则被限制。

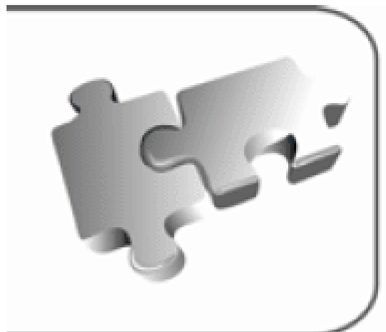
2. 利用 VBA 代码设置

将工作表 Sheet1 中的“B3：K10”单元格区域设置为允许滚动的区域，具体的操作步骤如下。

01 打开 VBA 代码窗口，在【工程资源管理器】窗口中选中〝Sheet1〞选项，在右侧代码窗口中输入如下程序代码。

```
Private Sub Workbook_Open()
  Worksheets(1).ScrollArea = " B3: K10"
End Sub
```

02 返回工作表，即可将〝B3：K10〞单元格区域设置为允许滚动的区域。当再次打开该工作簿时，也不必重新设置工作表的滚动区域。



9.9 综合实战 2——禁止查看指定工作表

对 Excel 文件中的一些重要内容，为防止他人查看或更改，用户可以将其所有的工作表设置为禁止查看的工作表。

将已有工作簿中的“财务明细表”设置为禁止查看，以确保“财务明细表”的信息安全，具体操作步骤如下。

01 在 VBA 左侧工程列表中选择要禁止查看的工作表，双击打开其代码窗口，并输入以下禁止查看〝财务明细表〞工作表的代码命令，如图 9-18 所示。

```
Private Sub Worksheet_Activate()
If ActiveSheet.Name Like "财务明细表" Then
'如果当前活动工作表的名称为“财务明细表”，则执行下一条语句。
 Sheets("加班明细表").Select
'当活动工作表为“财务明细表”工作表时，选中工作表“加班明细表”
 End If
End Sub
```

02 运行上述程序后，返回工作表中，先选择除工作表〝财务明细表〞之外的任意一个工作表，然后当再次选中工作表〝财务明细表〞时，即可驱动事件程序的运行，无法选中该工作表。

代码说明：Worksheet_Activate：当工作表成为活动工作表时，驱动事件程序运行。

Like 为字符串比较运算符，当字符串匹配样本时返回 True。



图 9-18　禁止查看“财务明细表”工作表程序代码



9.10 综合实战 3——指定区域密码保护

在日常使用中，有时需要对某个工作表中的某个区域进行保护。

保护工作表中 A 列的所有单元格，当选择 A 列单元格进行编辑时会自动弹出保护密码输入对话框，以验证是否有编辑权限，具体操作命令如下。

在 VBA 代码窗口中输入如下程序代码，如图 9-19 所示。

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
Dim pass
  If Target.Columns.Count > 1 Then
'判断选择的是否为 B 列，如果不是则维持原来的选择动作，即 ActiveCell.Select
      ActiveCell.Select
  Else
    If Target.Column = 1 And Columns("A:A").Locked = True Then
'判断如果是 A 列并且单元格已锁定，执行下面的语句。这里判断 Columns("A:A").Locked=True 的目的是防止选择 A 列单元格时不停的弹出输入密码提示框
        pass = InputBox("欢迎访问，请输入第 A 列保护密码", "录入提示")
'把 InputBox 对话框中输入的密码赋予变量 pass
        If pass = 1 Then
'如果输入的解除保护密码正确，则运行解除工作表保护(ActiveSheet.Unprotect Password:=123456)和取消 A 列锁定（Columns("A:A").Locked = False）的语句，最后再保护工作表（Sheets("第 1 节").Protect Password:=111111）
        ActiveSheet.Unprotect Password:=111111
        Columns("A:A").Locked = False
        Sheets("第 1 节").Protect Password:=111111
        End If
     End If
   End If
```

```
End Sub
```

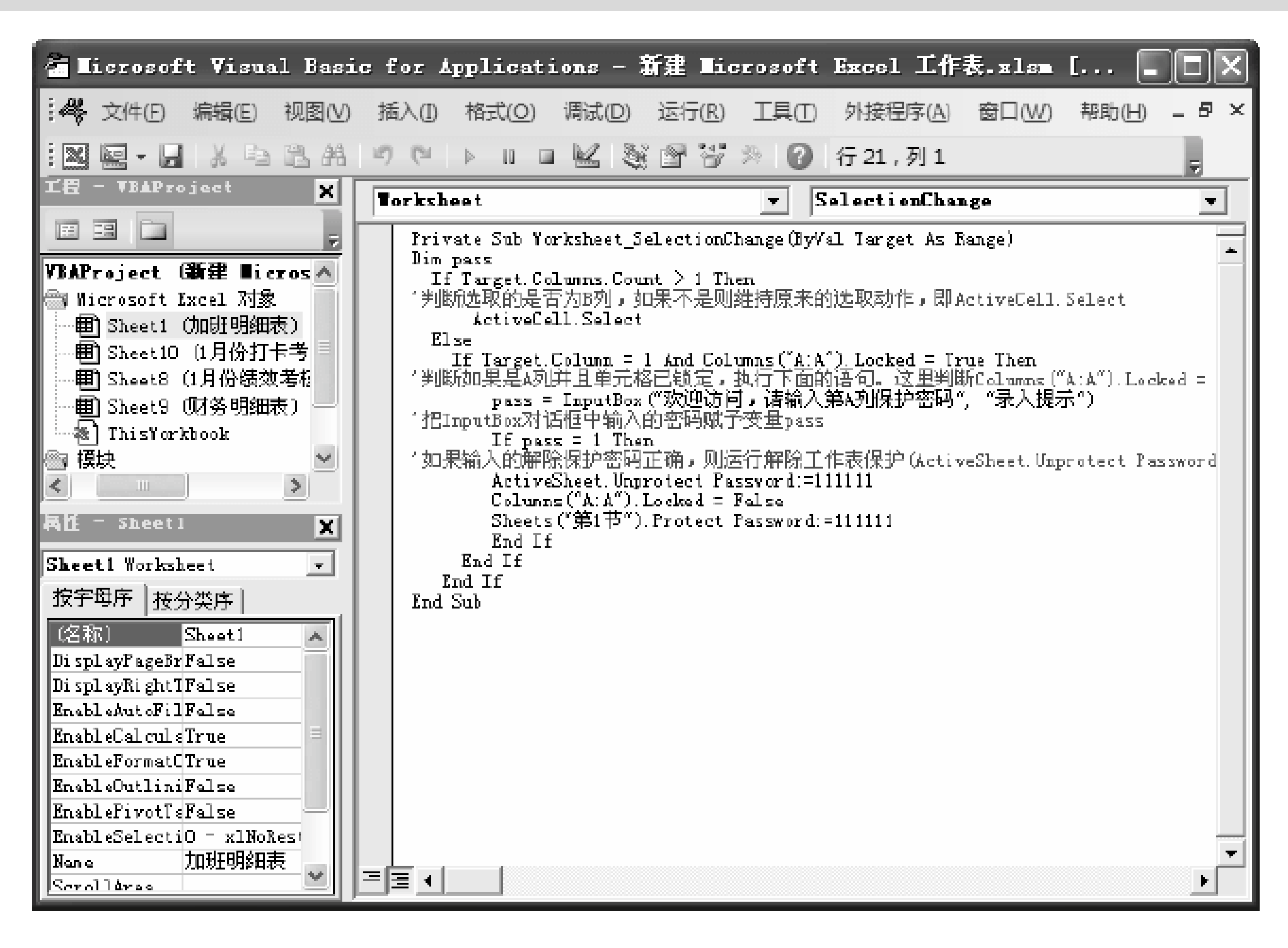


图 9-19　为指定区域设置密码保护

【提示】Worksheet_SelectionChange 是指当工作表内单元格发生选择改变时驱动程序运行的事件。

运行该程序后，当选择 A 列单元格时，系统则弹出如图 9-20 所示的输入密码提示对话框。

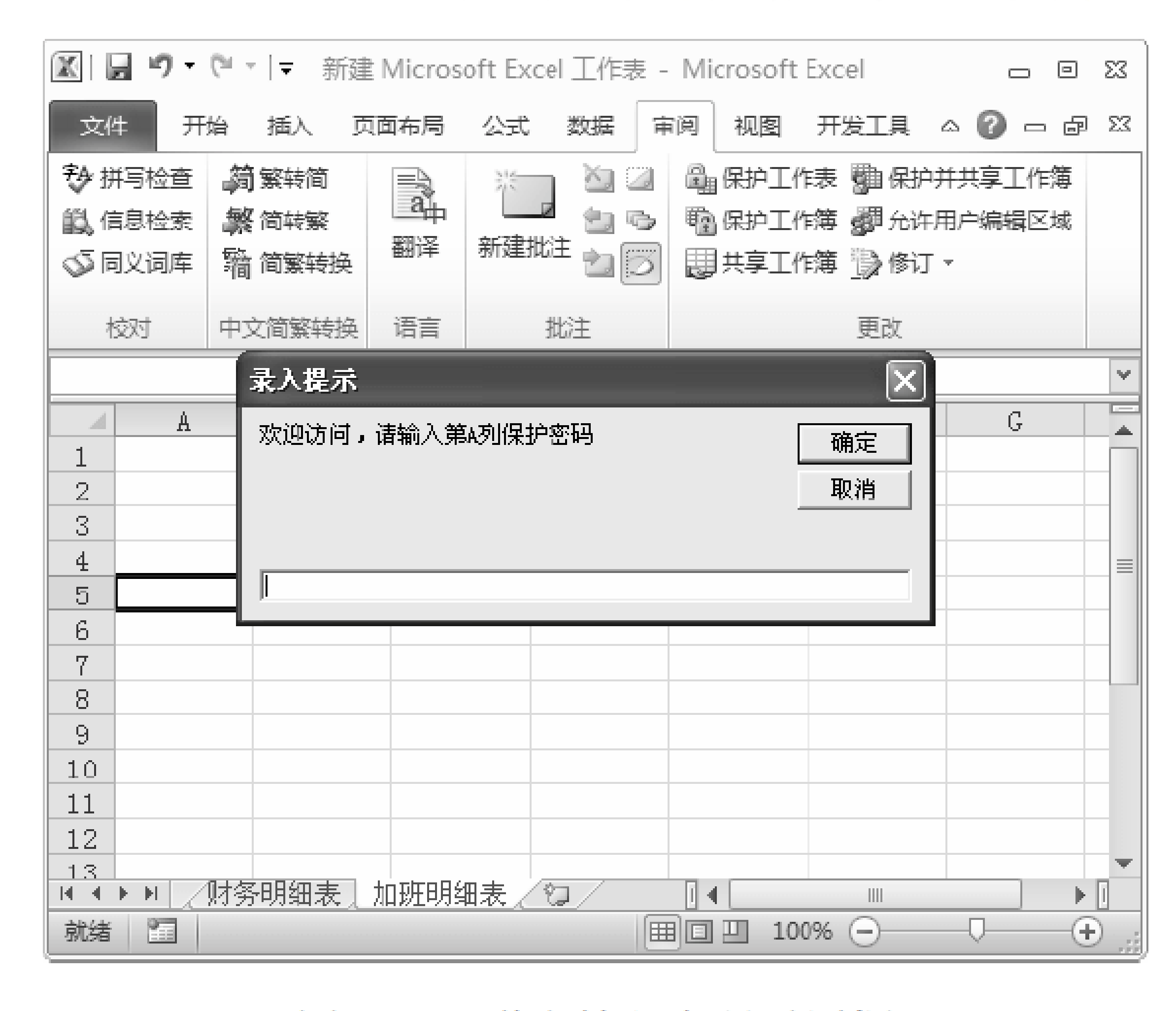


图 9-20　弹出输入密码对话框



9.11 高手私房菜

技巧 1：快速选择默认的工作表

如果要快速地选中默认工作表（如“Sheet1”），则可通过以下命令进行操作。

```
Sub 选中指定工作表()
  Sheet1.Select
End Sub
```

技巧 2：禁止随意修改工作表的名称

在 Excel 中经常会用到函数调用工作表名，所以一般禁止随意更改已规划好的工作表，下面就来介绍禁止更改工作表名称的方法。

例如，禁止更改工作簿中的“考勤表”工作表，具体 VBA 代码如下。

在 Excel 的基本操作中，可以通过保护工作簿的方法来禁止工作表的名称被更改。

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
  If Me.Name <> "考勤表" Then
    Me.Name = "考勤表"
    MsgBox "您好，该工作表名称禁止更改！", vbExclamation
  End If
End Sub
```

技巧 3：将工作表置为活动工作表

在使用 Excel 时，如果想要将其他特定表定义为当前的活动工作表，可以利用以下 VBA 代码来完成。

将工作表“工资表”定义为当前活动工作表，具体操作命令如下。

```
Sub 设置活动工作表()
  Worksheets("工资表").Activate
End Sub
```

其中 Activate 是工作表对象的方法，用于将工作表置于活动状态。

如果工作表使用的是默认工作表名（如“Sheet1”），可以使用以下命令使其成为活动工作表。

```
Sub 设置活动工作表()
  Sheet1.Activate
End Sub
```

使用 Select 和 Activate 操作同一张工作表时，二者实现的功能是一样的，都是将单个工作表选中或激活。它们的区别在于，Select 可以同时选择多张工作表，而 Activate 只能设置一张工作表为活动状态。

第 10 章　Excel 单元格相关操作

通过本章的学习，可以使读者掌握 Excel 单元格的相关操作，如单元格的表示、选择、输入、引用及删除等内容。这些内容在编程过程中会经常用到，熟练掌握单元格操作方法对一个 Excel VBA 使用者来说至关重要。



10.1 单元格的选择

单元格的选择是操作单元格的前提，其中包括选择全部单元格、选择单个单元格或选择单元格区域等。

10.1.1 选择单个单元格

选择单个单元格是最常用的，在 Excel VBA 中常使用 Range 和 Cells 来表示单元格，另外还可以使用一种简化的方法来表示单元格，下面分别对其介绍。

1. Range 表示单个单元格

Range 可以代表工作表中的某一个单元格、某一行、某一列或某一个选定区域等。

（1）使用 Range（单元格地址）的形式来表示某个单元格。例如选中单元格 A5，则可表示为：

```
Range("A5").Select
```

这里要注意，单元格要加双引号。

（2）使用 Range（"列标"&行数）的形式表示，如单元格 A5 表示为：

```
Range("A"&"5").Select
```

这里要注意，列标要加双引号，且要用“&”把列标和行号连接起来。

（3）如果需要选中定义名称的单元格或单元格区域，则可使用 Range（"定义的名称"）的形式来表示，并且要注意其中双引号的使用。如表示已定义名称为“月份”的单元格：

```
Range("月份")
```

2. Cells 表示单个单元格

Cells（row,column）代表单个单元格，其中 row 为行号，column 为列号数。

【例 10.1】使用 Cells 表示单个单元格 A5，具体操作方法如下。
使用 Cells 表示单个单元格的方式有以下 3 种。

（1）因为单元格的行数为 3，列数为 2，所以用以下命令可以实现。

```
Cells(3,2)
```

（2）使用“Cells（行号，列标）”的表示方法，命令如下。

```
Cells(3, "B")
```

（3）使用“Cells（单元格序号）”的表示方法，命令如下。

```
Cells(514)
```

【提示】对于 Excel 工作表中的 256（列）*65536（行）个单元格，每个单元格都有其对应的序号。具体的对应关系是，单元格序号=（行号-1）*256+列号。

根据单元格的对应序号的计算方法，在工作表的前 20 行中显示当前单元格序号，其程序代码如下：

```
Sub 显示单元格序号()
  For I = 1 To 256
     For J = 1 To 20
       Cells(J, I) = (J - 1) * 256 + I
    Next J
  Next I
End Sub
```

运行上述程序后，即可在指定单元格区域内显示其对应的单元格序号，如图 10-1 所示。

	A	B	C	D	E	F	G	H
1	1	2	3	4	5	6	7	
2	257	258	259	260	261	262	263	26
3	513	514	515	516	517	518	519	52
4	769	770	771	772	773	774	775	77
5	1025	1026	1027	1028	1029	1030	1031	103
6	1281	1282	1283	1284	1285	1286	1287	128
7	1537	1538	1539	1540	1541	1542	1543	154
8	1793	1794	1795	1796	1797	1798	1799	180
9	2049	2050	2051	2052	2053	2054	2055	205
10	2305	2306	2307	2308	2309	2310	2311	231
11	2561	2562	2563	2564	2565	2566	2567	256
12	2817	2818	2819	2820	2821	2822	2823	282
13	3073	3074	3075	3076	3077	3078	3079	308
14	3329	3330	3331	3332	3333	3334	3335	333
15	3585	3586	3587	3588	3589	3590	3591	35

图 10-1　显示当前单元格序号

3. 单元格表示的简化方法

除了使用 Range 和 Cell 来表示单元格外，还可以使用一些简化的方法来表示单元格。单元格的简化表示方式为［单元格地址］。

例如，选中当前工作表中的单元格 A5，则可使用如下代码：

```
Sub 单元格简化方法()
   [A5].Select
End Sub
```

10.1.2 选择全部单元格

在工作表中，可以单击 Excel 工作表中“行号”和“列标”交汇处的【全选】按钮，或按 Ctrl+A 组合键来选择工作表中所有的单元格，如图 10-2 所示。

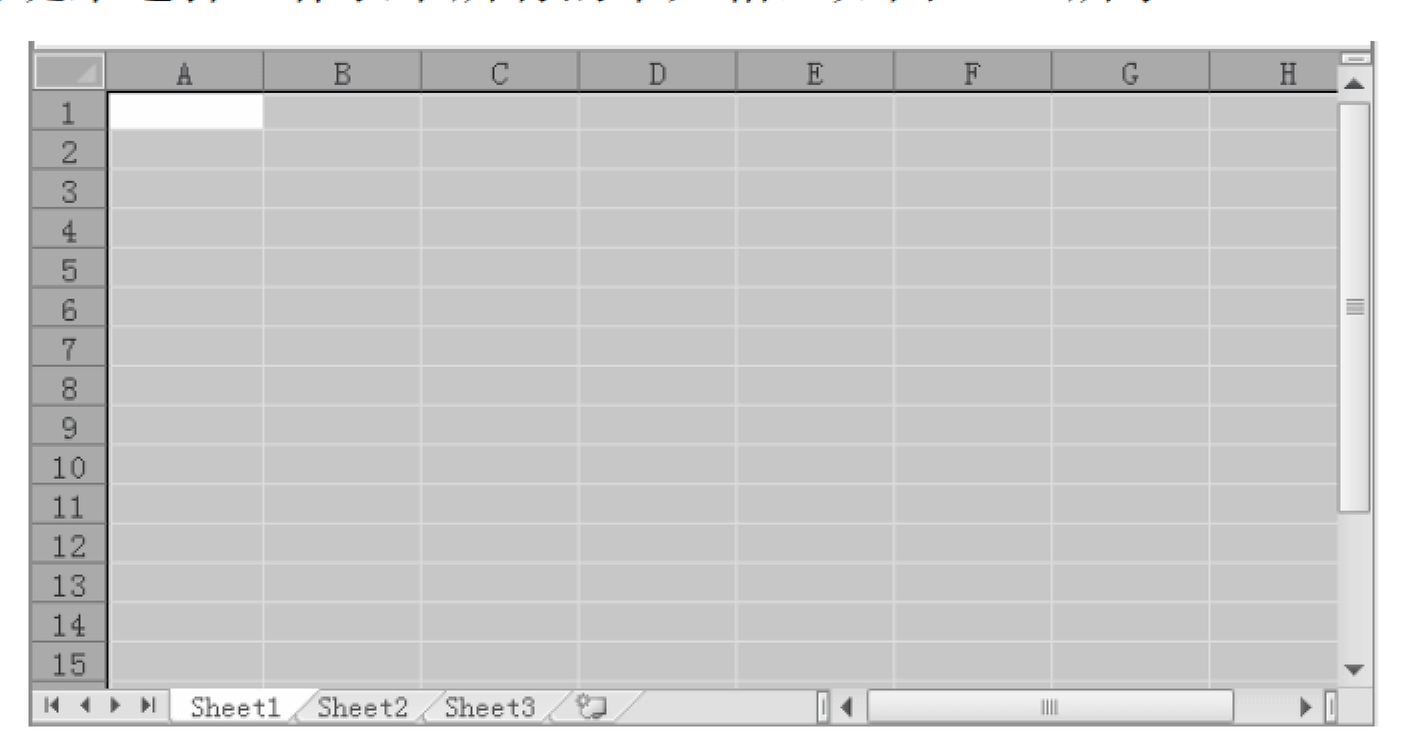


图 10-2　全选工作表所有单元格

如果要利用 Excel VBA 语句来完成选择工作表中的全部单元格，可以通过以下代码来实现。

```
Sub 选择全部单元格 1()
   Cells.Select
End Sub
```

其中 Cells 是所有单元格的集合。

除此之外，还可以使用以下两种代码选择工作表的全部单元格。

```
Sub 选择全部单元格 2()
   Rows.Select
End Sub
```

或

```
Sub 选择全部单元格 3()
   Columns.Select
End Sub
```

其中 Rows 和 Columns 分别是所有行和列的集合。



10.1.3 选择指定的单元格

当需要表示具体某个工作表中的单元格时，可以在单元格表示方法前添加具体的工作表名称，并用“.”来表示从属关系，命令格式如下：

```
Worksheets("工作表名称").Range("单元格地址")
```

如果需要表示某个工作簿中某个工作表中的某个单元格，则可再添加上工作簿的名称。

```
Workbook("工作簿的名称").Worksheets("工作表名称").Range ("单元格地址")
```

除了以上方法外，还可以先用命令将指定工作表或者工作簿设置为当前活动工作簿或工作表，然后再执行单元格选择操作。

【例 10.2】选择非活动工作表“考勤表”的 A5 单元格，具体操作命令如下。

```
Sub 选择非活动工作表单元格 A5()
   Sheets("考勤表").Select
   Range("A5").Select
End Sub
```

【提示】使用上述方式时必须先将工作表设置为当前活动工作表，然后再选择单元格，不能够同时执行活动工作表和选择单元格的操作，以下程序代码为错误命令格式，执行后弹出如图 10-3 所示的错误提示信息。

```
Sub 选择非活动工作表单元格 A5()
   Sheets("考勤表"). Range("A5").Select
End Sub
```



图 10-3　错误提示信息

10.1.4 选择单元格区域

在掌握了用 Excel VBA 选择单个单元格后，接下来介绍如何选择多个单元格区域。选择多个单元格区域分为选择连续单元格区域和不连续单元格区域。

1. 选择连续单元格区域

【例 10.3】选择当前活动工作表的 A1:F10 单元格区域，如图 10-4 所示，具体操作方法如下。

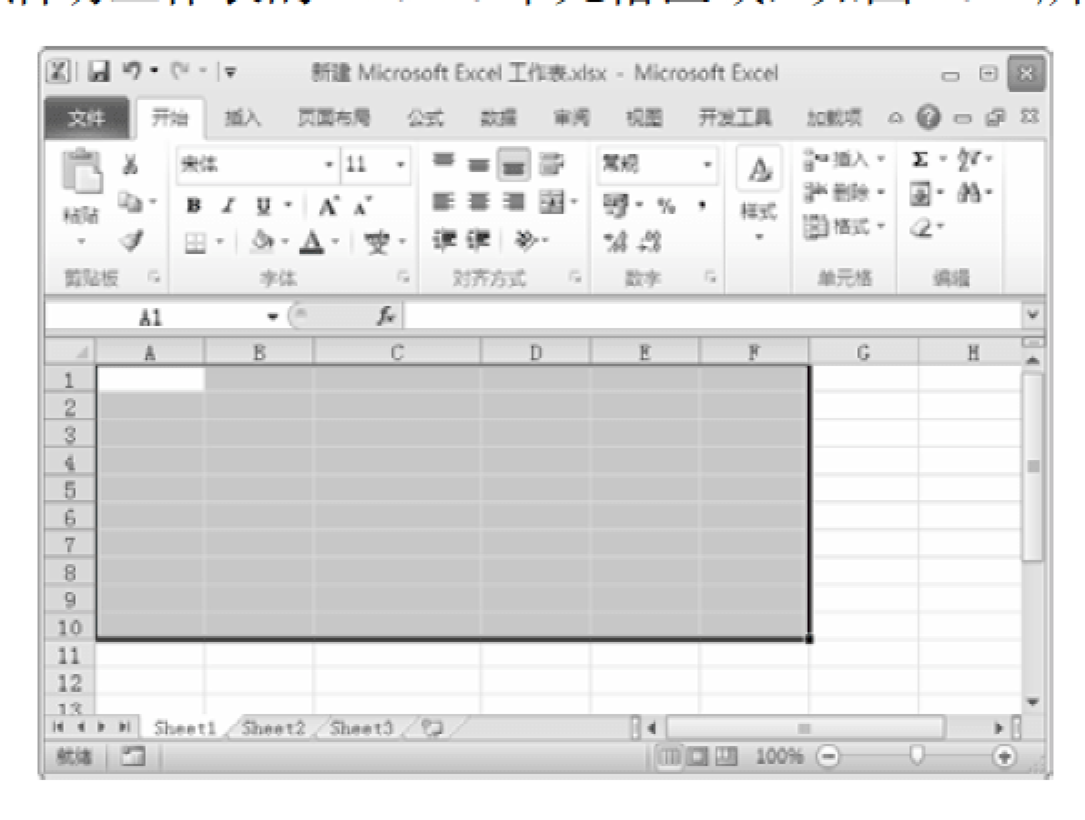


图 10-4　选择连续区域

可以使用多种程序代码实现单元格区域的选择，介绍如下。

（1）利用 Range（"单元格区域"）的形式，命令如下。

```
Sub 选择连续单元格方法 1()
   Range("A1:F10").Select
End Sub
```

（2）利用 Range（"区域的左上角单元格"，"区域的右下角单元格"）的形式，命令如下。

```
Sub 选择连续单元格方法 2()
   Range("A1", "F10").Select
End Sub
```

（3）利用 Range 和 Cells 结合的形式，命令如下。

```
Sub 选择连续单元格方法 3()
   Range(Cells(1,1), Cells(10,6)).Select
End Sub
```

【提示】 Range(Cells(1,1), Cells(10,6))是 Cells 方法用来分别表示区域的左上角和右下角单元格的。

（4）利用中括号“[]”简化单元格区域表示，命令如下。

```
Sub 选择连续区域方法 4()
   [A1:F10].Select
End Sub
```

在工作表中，可以通过选择行号或列标的方式来选择一行或多行、一列或多列，可以使用 Range 来表示行和列，也可以直接使用 Rows 属性和 Columns 属性来表示。

【例 10.4】选中工作表中的第 5 行，具体操作命令如下。

```
Sub 选中第 5 行()
```

```
    Range("5:5").Select
End Sub
```

【技巧】其中 Range("5:5").Select 可以使用 Rows("5:5").Select 或 Rows(5).Select 替换。

【例 10.5】选中工作表中的第 5 行至第 10 行，具体操作命令如下。

```
Sub 选中第 5 至 10 行 1()
    Range("5:10").Select
End Sub
```

【技巧】其中 Range("5:10").Select 可用 Rows("5:10").Select 替换。

【例 10.6】选中工作表中的 B~E 列，具体操作命令如下。

```
Sub 选中工作表 B 到 E 列 ()
        Columns("B:E").Select
End Sub
```

【技巧】如果只选择一列，如 B 列，可以写成 Columns("B:B").Select 或者 Columns(2).Select。

选中多列的方式和选择多行的方式相似，这里就不再介绍。

2. 选择不连续单元格区域

不连续区域是不相邻的单元格或单元格区域。在 VBA 中可以使用 Range 对象来表示不连续的区域，也可以使用 Union 函数来完成多个不连续区域的选择。具体内容介绍如下。

（1）利用 Range 对象表示不连续区域。

【例 10.7】选择不连续单元格 A1,A3:D3,A6:F6,A10:H10，具体操作命令如下。

```
Sub 不连续区域()
    Range("A1,A3:D3,A6:F6,A10:H10").Select
End Sub
```

【提示】利用 Range 表示不连续区域的时候，单元格或单元格区域之间应用“,”（英文输入法状态）分隔。

运行上述代码，即可选中不连续区域的单元格区域，如图 10-5 所示。

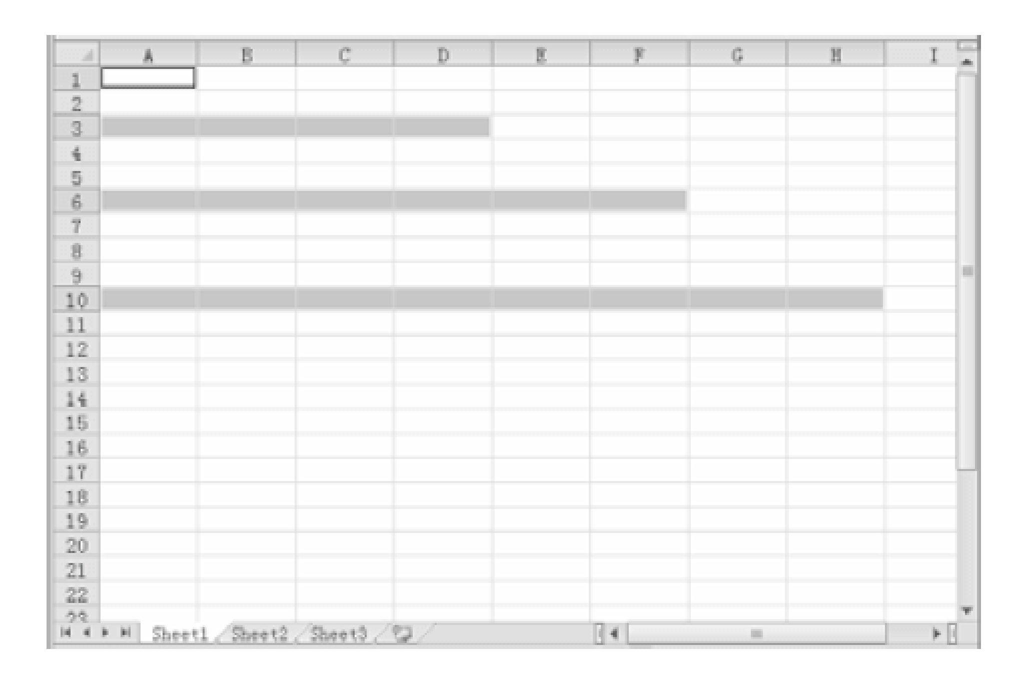


图 10-5　使用 Range 选择不连续单元格区域

如果是选择不连续的行或列，也可以用 Range 对象来实现，程序代码如下。

```
Sub 不连续行()
   Range("3:3,5:6,9:12").Select
End Sub
```

运行上述程序后，返回工作表中，会看到同时选中工作表的第 3 行、第 5 行、第 6 行和第 9~12 行，如图 10-6 所示。

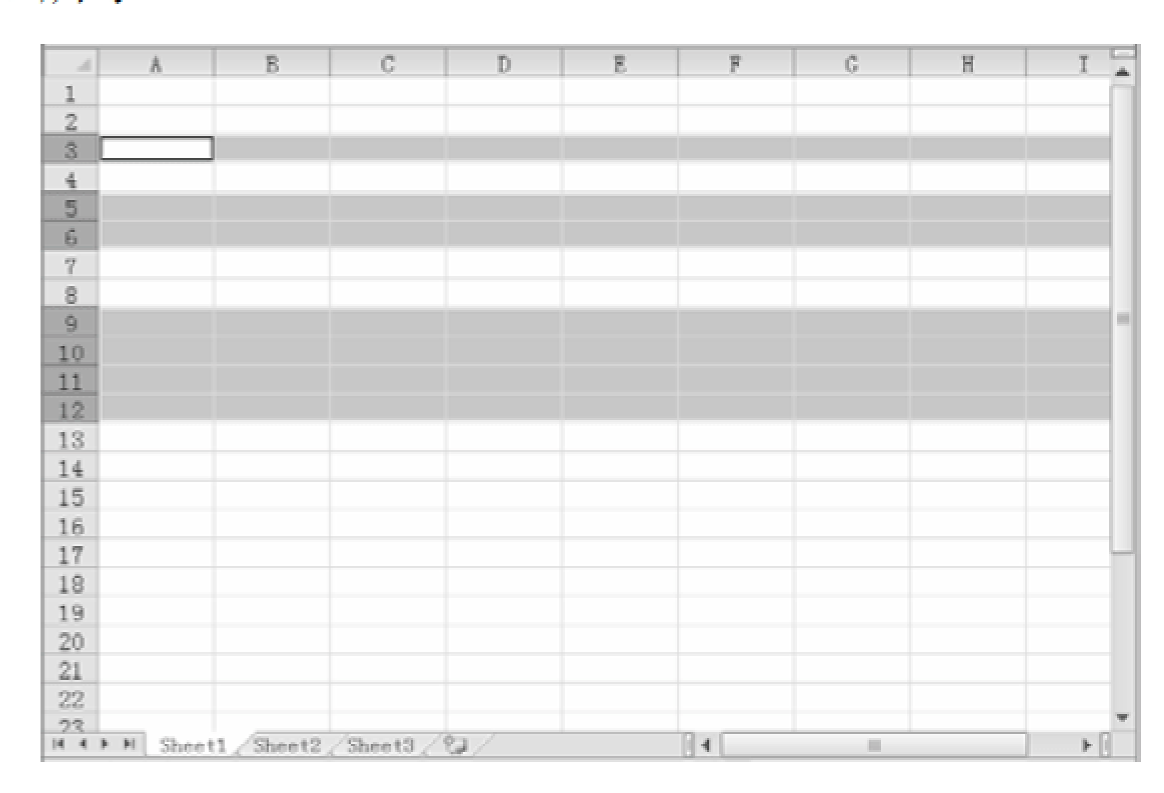


图 10-6 使用 Range 选择不连续的行

【提示】 不能用 Rows("3:3,5:6,9:12")来代替 Range("3:3,5:6,9:12")，否则在运行程序时会弹出如图 10-7 所示的错误提示信息。

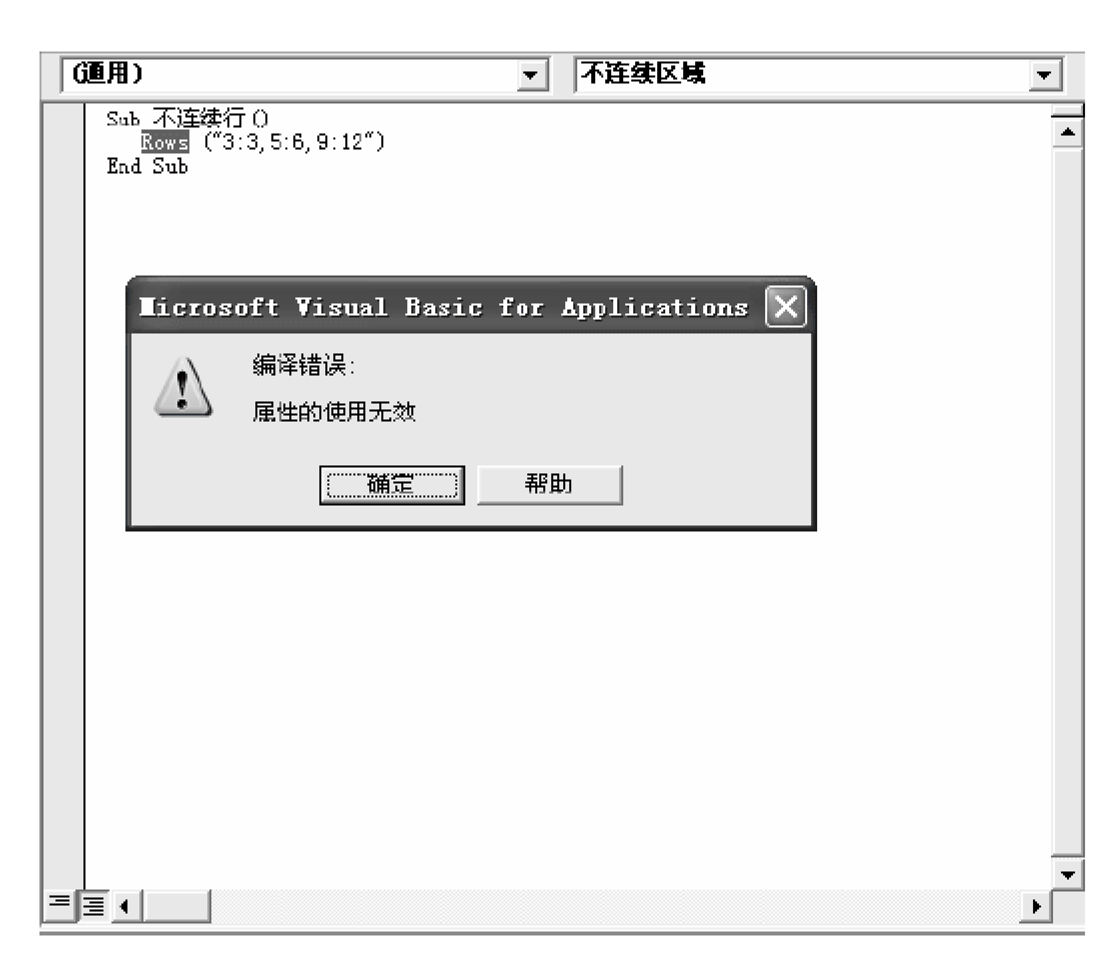


图 10-7 错误提示信息

当需要选择不连续的列时，例如选择 C 列、第 F 至 H 列和 J 列，其程序代码如下。

```
Sub 选择不连续列()
   Range("C:C,F:H, J:J").Select
End Sub
```

运行上述程序后，即可选择 C 列、第 F 至 H 列和 J 列不连续的单元格区域，如图 10-8 所示。

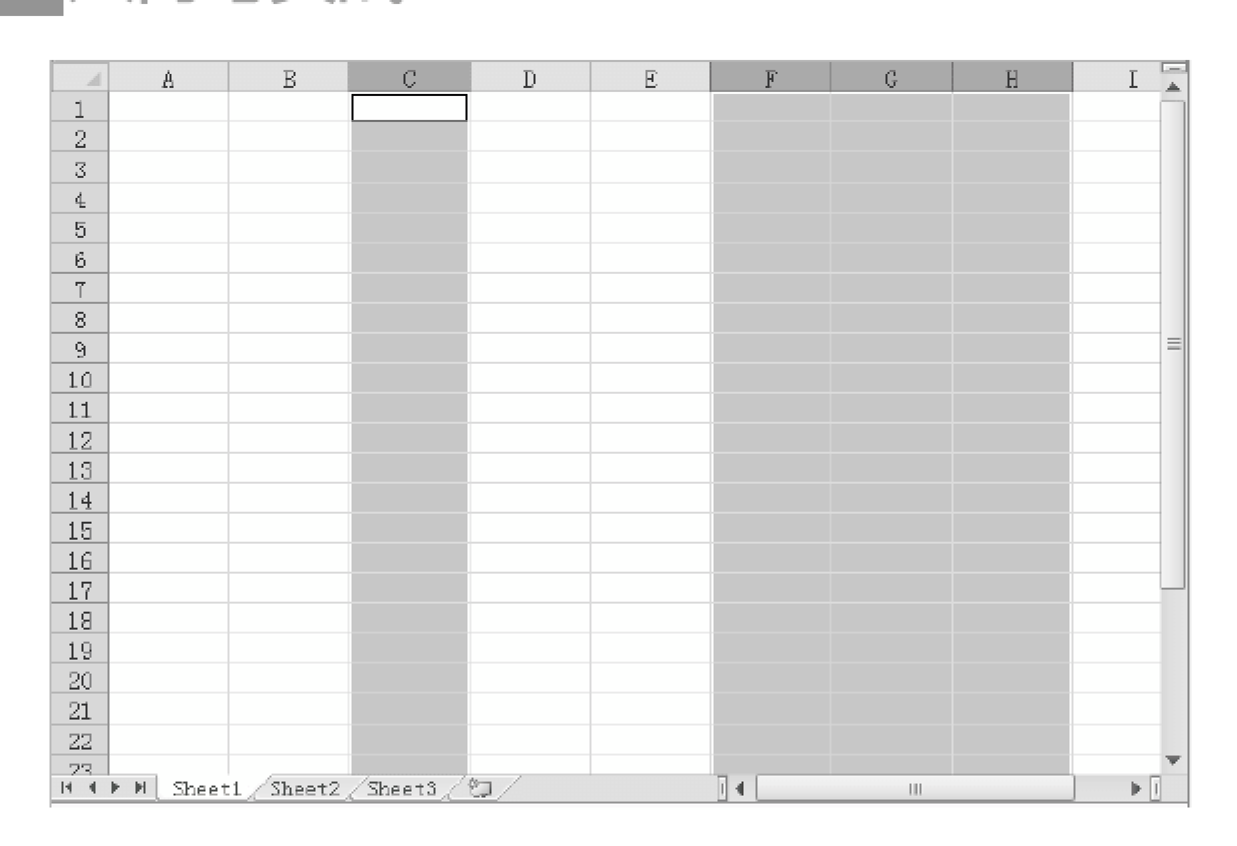


图 10-8　选择不连续的列单元格区域

【提示】这里同样不可以使用 Columns("C:C,F:H, J:J")来代替 Range("C:C,F:H, J:J")。

（2）使用 Union 函数选择不连续区域。

如果使用 Union 函数，则是将两个或两个以上的区域组合成一个范围。例如，使用 Union 函数选择和上述代码同样的不连续区域，其代码如下。

```
Sub 不连续区域2()
  Union(Range("A1"), Range("A3:D3"), Range("A6:F6"), Range("A10:H10")).Select
End Sub
```

运行上述程序代码后，同样可以选择如图 10-5 所示的不连续单元格区域。

10.1.5 选择特定单元格

Excel 的定位功能可以选择特定单元格，如选择空值、选择可见单元格等。

1. 选择当前已使用的单元格

使用工作表的 UsedRange 属性可以选择当前已使用的单元格。当前已使用的单元格是指工作表中已经使用过的单元格。无论是否存有数据，只要曾经使用过都属于当前已使用的单元格。

如图 10-9 所示的工作表中，C5:H19 区域为已使用的单元格区域。要选择该区域，其程序代码如下。

```
Sub 当前已使用区域()
   ActiveSheet.UsedRange.Select
End Sub
```

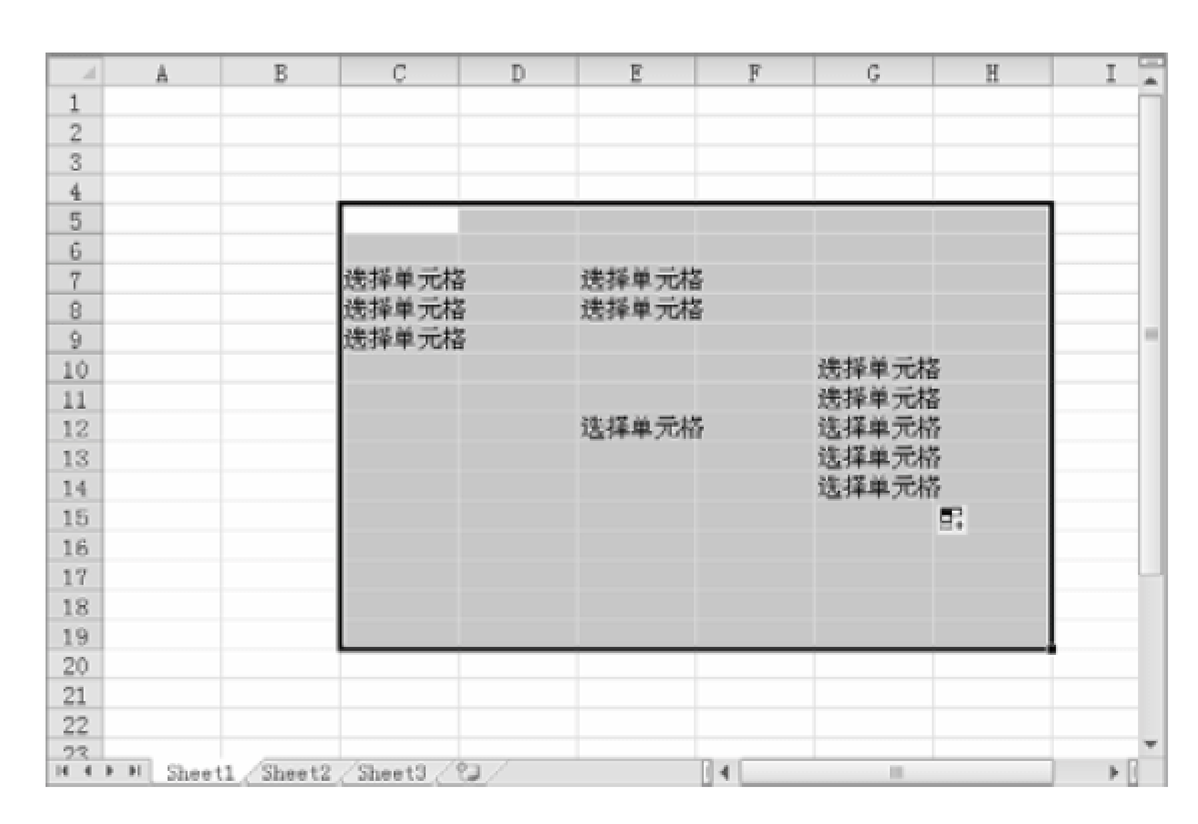


图 10-9　选择工作表中已使用的区域

【提示】ActiveSheet 是指当前活动工作表；ActiveSheet.UsedRange 是指当前活动工作表中已使用的单元格；UsedRange 属性只应用于工作表对象（Worksheet），它的前面不能是工作簿对象（workbook）或单元格对象（Range）。

运行上述程序，即可选择当前已使用的单元格区域。

2. 选择和指定单元格相邻的区域

在选择一个含有数据的动态区域时，可以利用判断语句对单元格逐个进行判断再选择，但程序会变得复杂起来。要使选择相邻区域变得灵活，可以使用下面的语句。

```
Range("单元格区域").CurrentRegion.Select
```

使用下面的语句，可以快速选择相邻的连续区域：

```
Sub 选中相邻的连续区域()
   Range("D8").CurrentRegion.Select
End Sub
```

【提示】Range("D8").CurrentRegion 表示以 D8 为起点的相邻区域。

运行上述程序，即可选择相邻的连续单元格区域，如图 10-10 所示。

图 10-10　选择工作表中相邻区域

【提示】相邻区域是指和已有连续区域的行和列都相邻的区域。

3. 选择区域端点单元格

在一个列区域内，想要选择其中最上方的第一个非空单元格或最下方非空单元格，可以使用“Ctrl+方向键”的方式实现。例如，在如图 10-11 所示的工作表中，如果要快速选择 F 列的第一个非空单元格 F11，可以在 F1 中使用 Ctrl+↓（向下箭头）来选择；要选择数据区域右边的端点单元格 G11，可以在 F11 中按 Ctrl+→（向右箭头）来选择。

以上功能在 Excel VBA 中，利用 Range 对象的 End 属性来实现，具体操作方法如下。

例如，在如图 10-11 中选择 F 列中第一个非空单元格，其程序代码如下：

```
Sub F列第一个非空单元格()
    If Range("F1") = "" Then
        Range("F1").End(xlDown).Select
    Else
        Range("F1").Select
    End If
End Sub
```

【提示】If Range("F1") = ""的功能是，如果 F1 的值为空；Range("F1").End(xlDown)相当于在 F1 中按 Ctrl+↓（向下箭头）。

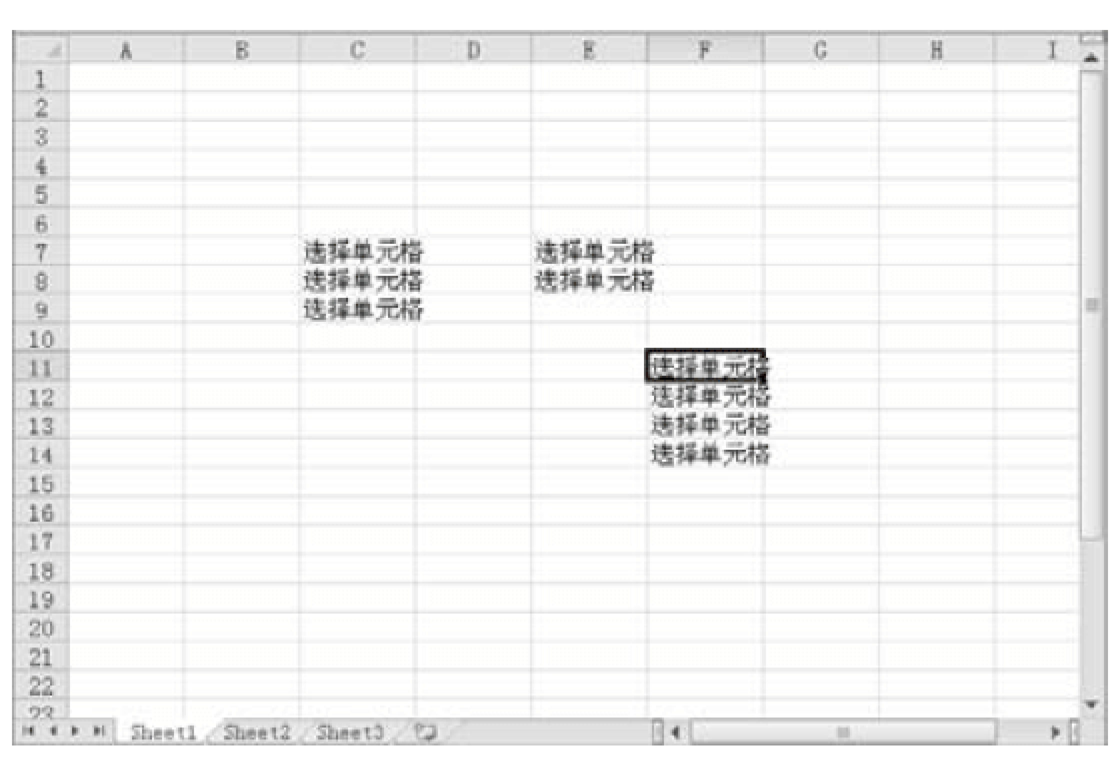


图 10-11　选择 F 列第一个非空单元格

【提示】End 参数与方向键对照关系如下。

- End(xlup)相当于 Ctrl+↑（向上箭头）。
- End(xldown)相当于 Ctrl+↓（向下箭头）。
- End(xltoleft)相当于 Ctrl+←（向左箭头）。
- End(xltoright)相当于 Ctrl+→（向右箭头）。

如果要选择 F 列最后一个非空单元格，其程序代码如下：

```
Sub 选择F列最后一个非空单元格()
    Range("F2565").End(xlUp).Select
End Sub
```

【提示】Range("F2565").End(xlUp)：F 列最后一个非空单元格需要从 F 列最后一个单元格向上查找。

运行上述程序后，即可选择 F 列数据区域中最后一个非空单元格 F14。

如果要选择 F 列数据区域下的第一个空白单元格，可以使用以下程序代码：

```
Sub 数据区域下第一个空白单元格()
    Dim i As Integer
    i = Range("F2821").End(xlUp).Row + 1
        ' 该空白单元格的行数等于 F 列最后一个非空单元格的行数加 1，Row 是指单元格的行数
    Range("E" & i).Select
End Sub
```

运行上述程序后，即可选择 F 列数据区域下的第一个空白单元格 F15。

如果要选择第 9 行最左边一个非空单元格，可使用如下程序代码：

```
Sub 选择第 9 行最左边一个非空单元格()
    If Range("A9") = "" Then
        Range("A9").End(xlToRight).Select
    Else
        Range("A9").Select
    End If
End Sub
```

运行上述程序后，即可选择第 9 行最左边一个非空单元格 C9。

若要选择第 9 行最右边一个非空单元格，可以使用下面的程序代码：

```
Sub 选择第 9 行最右边一个非空单元格()
    Range("IV9").End(xlToLeft).Select
End Sub
```

运行上述程序，即可选择工作表中第 9 行最右边的一个非空单元格 C9（第 9 行只有一个单元格有数据）。

4. 选择特定内容单元格

当需要选择如常量、公式等特定单元格时，可以选择功能区【开始】➤【查找和选择】➤【定位条件】菜单命令，打开【定位条件】对话框，如图 10-12 所示，从中选择要定位的条件，就可以选择特定的单元格。

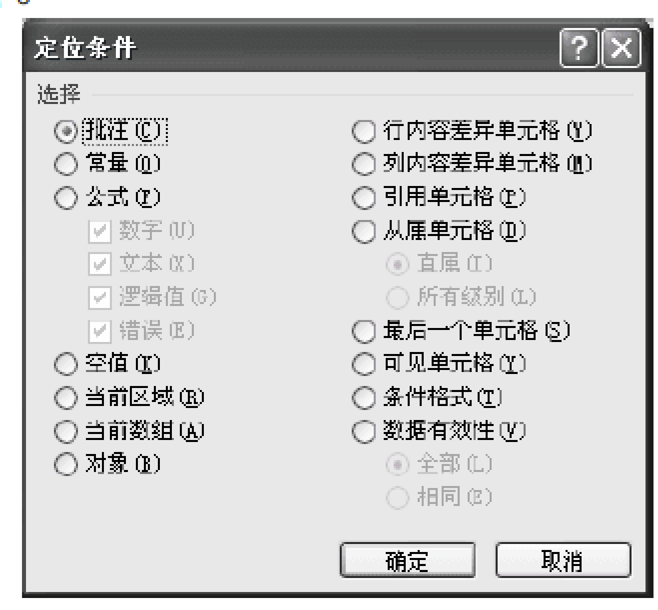


图 10-12　【定位条件】对话框



如果在 Excel VBA 中需要选择特定的单元格，可以使用 Range 对象的 SpecialCells 方法来实现。选择特定单元格的格式如下：

```
Range("单元格区域").SpecialCells(单元格类型变量)
```

单元格类型常量的详细说明如表 10-1 所示。

表 10-1　单元格类型常量的详细说明

常　量	说　明
xlCellTypeAllValidation	具有有效条件的单元格
xlCellTypeBlanks	空单元格
xlCellTypeConstants	包含常量的单元格
xlCellTypeFormulas	包含公式的单元格
xlCellTypeComments	包含注释的单元格
xlCellTypeSameFormatConditions	具有相同格式的单元格
xlCellTypeSameValidation	具有相同有效条件的单元格
xlCellTypeLastCell	已用区域的最后一个单元格
xlCellTypeVisible	所有可见单元格
xlCellTypeAllFormatConditions	任意格式的单元格

在实际工作中，使用 SpecialCells 命令，可以提高工作效率，例如删除图 10-13 所示工作表中指定数据区域 C3：C11 中单元格为空的行，具体程序代码如下。

```
Sub 删除单元格为空的行()
   Range("C3:C11").SpecialCells(xlCellTypeBlanks).EntireRow.Delete
End Sub
```

运行上述程序后，即可将“销售量”为空的单元格所在的行删除，如图 10-14 所示。



图 10-13　商品销售统计表

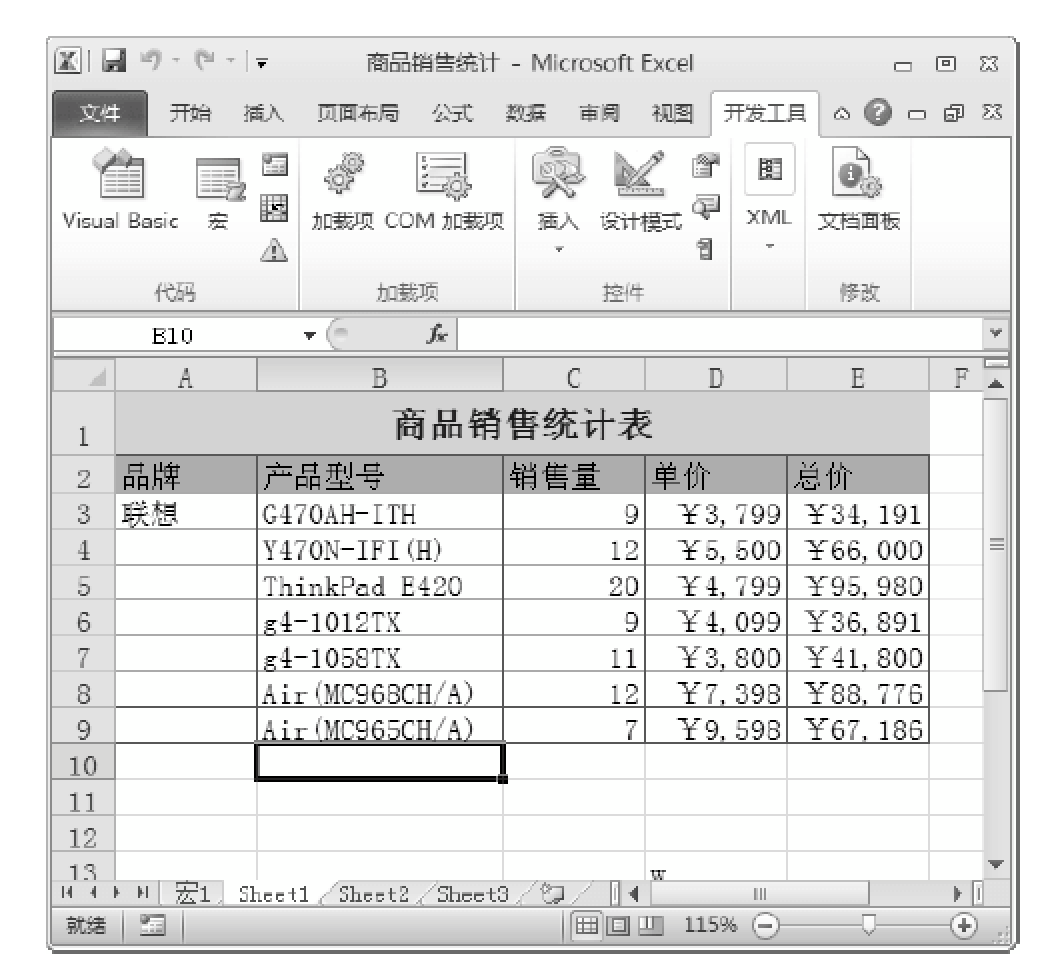


图 10-14　删除“销售量”为空的行

10.1.6 移动和改变单元格选择范围

移动和改变单元格的选择范围是 Excel 应用的基础。下面来介绍一下使用 VBA 移动和改变单元格选择范围的方法。

在 Excel 中主要使用 Offset 函数实现单元格的移动和范围选择。

【例 10.7】以 D4 单元格为参照，向下移动 3 行，向左移动 2 列，并选择 4 行 9 列的单元格区域，具体操作命令如下。

```
=Offset(D4,3,2,4,9)
```

在 VBA 中，Offset 可以被拆分成两个独立的函数使用。

```
Offset（移动行数，移动列数）
```

和

```
Resize（总行数，总列数）
```

【例 10.8】使用 Offset 以 A2 为参考，移动选择 C6 单元格，具体操作命令如下。

```
Sub 移动选择()
   Range("A2").Offset(4, 2).Select
End Sub
```

执行命令后，会自动选择 C6 单元格，如图 10-15 所示。

图 10-15　相对位置的移动

【例 10.9】以 C6 为参照，选择行数为 6，列数为 3 的单元格区域，具体操作命令如下。

```
Sub 固定大小的范围选择()
   Range("C6").Resize(6, 3).Select
End Sub
```

运行该程序后，即可选择以 C6 为参照、行数为 6、列数为 3 的单元格区域，如图 10-16 所示。

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5			单元格				单元格	
6			单元格				单元格	
7			单元格		单元格		单元格	
8			单元格		单元格		单元格	
9			单元格		单元格		单元格	
10			单元格		单元格		单元格	
11			单元格		单元格		单元格	
12			单元格		单元格		单元格	
13			单元格		单元格		单元格	
14			单元格		单元格		单元格	
15			单元格		单元格		单元格	
16			单元格		单元格		单元格	
17					单元格			
18					单元格			
19								
20								

图 10-16　选择固定大小的范围

【提示】Offset 和 Resize 函数可以同时使用，如以 A2 为参照表示移动所在单元格区域，同样可以选择如图 10-16 所示的固定范围，具体操作命令如下。

```
Sub 移动并选择固定范围()
   Range("A2").Offset(4, 2).Resize(6, 3).Select
End Sub
```



10.2 对单元格进行赋值

为单元格赋值，填入数据内容是单元格操作中最基本的内容。单元格赋值的操作内容介绍如下。

10.2.1 直接赋值与引用

可以通过直接赋值和引用的方式为单元格填充数据内容。其中直接赋值可以利用单元格对象的 Value 属性来操作，其语法格式如下：

```
单元格.Value = ("常量")
```

【例 10.10】在工作表的“B2:E10”单元格中输入“你好”，具体操作命令如下。

```
Sub 直接赋值()
   Range("B2:E10").Value = "你好"
End Sub
```

运行该程序后，即可在当前工作表中的“B2:E10”单元格区域中输入“你好”，如图 10-17 所示。

B2 | f_x 你好

	A	B	C	D	E	F	G
1							
2		你好	你好	你好	你好		
3		你好	你好	你好	你好		
4		你好	你好	你好	你好		
5		你好	你好	你好	你好		
6		你好	你好	你好	你好		
7		你好	你好	你好	你好		
8		你好	你好	你好	你好		
9		你好	你好	你好	你好		
10		你好	你好	你好	你好		
11							
12							

图 10-17　直接赋值

【提示】在单元格中输入内容时，Value 可以省略，但是输入的常量必须添加双引号（" "）。也可以引用其他单元格中的值为特定区域的单元格赋值。

【例 10.11】在单元格区域 B13:D15 中引用单元格 B2 中的值，具体操作命令如下。

```
Sub 直接引用()
   Range("B13:D15") = Range("B2")
End Sub
```

运行该程序，为 B13:D15 区域的单元格赋予了和 B2 单元格相同的值，如图 10-18 所示。

D6 | f_x Hello

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6				Hello	Hello	Hello	Hello	
7				Hello	Hello	Hello	Hello	
8				Hello	Hello	Hello	Hello	
9				Hello	Hello	Hello	Hello	
10				Hello	Hello	Hello	Hello	
11				Hello	Hello	Hello	Hello	
12				Hello	Hello	Hello	Hello	
13				Hello	Hello	Hello	Hello	
14								
15								
16								
17				Hello	Hello	Hello	Hello	
18				Hello	Hello	Hello	Hello	
19				Hello	Hello	Hello	Hello	
20				Hello	Hello	Hello	Hello	
21								
22								

图 10-18　直接引用

10.2.3 利用公式赋值

除了简单地将数据赋予单元格外，还可以通过公式为单元格赋予运算结果，其中主要利用单元格的 Formula 属性来实现该功能。具体内容介绍如下。

【例 10.12】为单元格 D5 赋予 B5*C5 的结果，具体操作命令如下。

```
Sub 利用公式赋值()
   Range("D5").Formula = "=B5*C5"
End Sub
```

运行该程序，在 D5 单元格中显示输入相应公式的计算结果，如图 10-19 所示。

D5 =B5*C5

	A	B	C	D	E
1					
2					
3					
4					
5		4	5	20	
6					
7					
8					

图 10-19　利用公式赋值

【提示】在单元格中输入公式时，Formula 也可以省略，但是必须添加等号（=）和双引号（" "）。如果没有等号，则相当于输入了常量。

如果要查看单元格中的公式，可以利用 Formula 属性来实现，操作命令如下。

```
    Sub 获取单元格中的公式()
     MsgBox "单元格 D5 中的公式为：" & Range("D5").Formula, vbInformation, "获取公式
"
    End Sub
```

运行该程序，可弹出【获取公式】的提示信息，如图 10-20 所示。

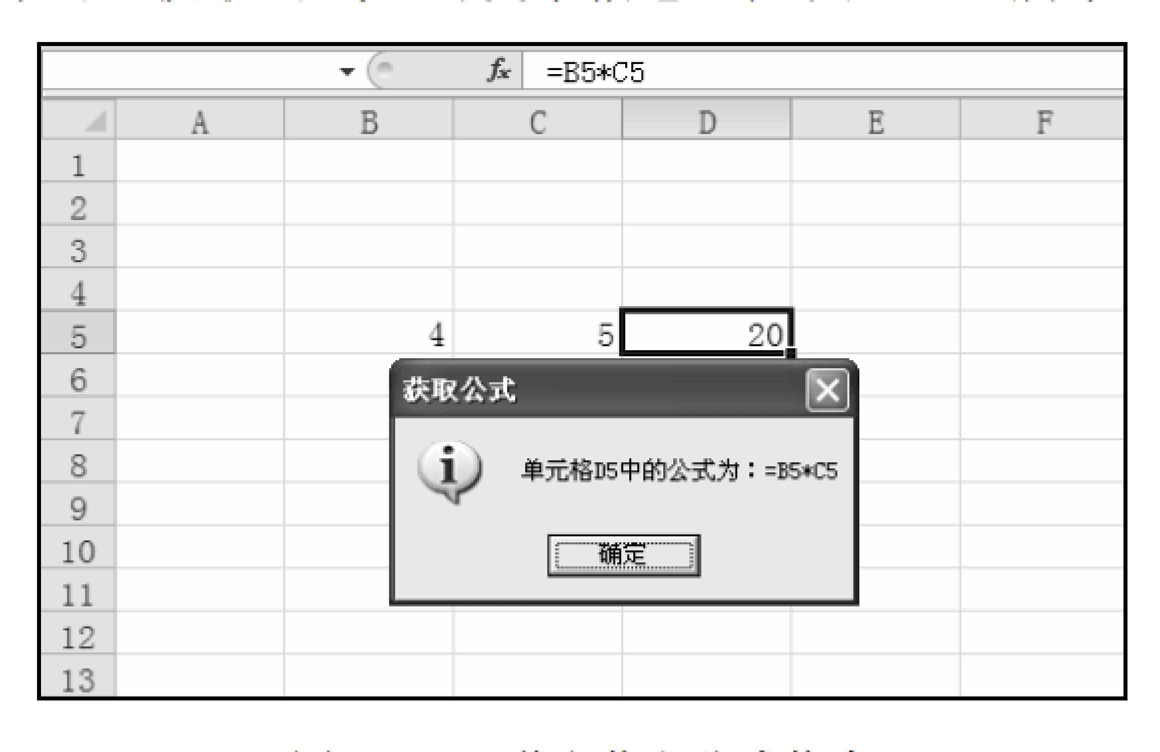


图 10-20　弹出获取公式信息

10.2.3 引用其他工作表中的单元格

在赋值时经常会遇到引用其他工作表中数据的现象，这时只需在被引用的单元格的前面加上“工作表名!”即可。

【例 10.13】在当前工作表 Sheet1 的 B5 单元格中，输入 Sheet2 工作表中 A1:A10 区域的数值和，具体操作命令如下。

```
    Sub 跨工作表引用()
            Range("B5").Formula = "=SUM(Sheet2!A1:A10)"
    End Sub
```

【提示】若被引用的工作表名称被重命名，并且该名称有可能和其他信息产生歧义的话，需要使用单引号“'”将引用的工作表的名称括起来，例如下面的语句：

```
Sub 跨工作表引用()
Range("B5").Formula = "=SUM( '工资表' !D2:D10)"
End Sub
```

10.2.4 引用其他工作簿中的单元格

除了跨工作表引用数据外，还经常有跨工作簿引用数据的现象，这时只需在被引用的单元格所在工作表名称的前面加上“[工作簿名]”即可。

【例 10.14】获取员工考勤工作簿中 Sheet2 表的 D3:D10 区域中的最小值，具体操作命令如下。

```
Sub 跨工作簿引用()
ActiveCell.Formula = "=MIN([员工考勤.xlsx]Sheet2!D3:D10)"
End Sub
```

【提示】以上命令所引用的工作簿没有指明绝对路径，默认为桌面上的工作簿，如果需要引用不在桌面的工作簿，需要在工作簿名称前加上绝对路径。例如，引用“D:\第一季度考勤\”目录下的工作簿，操作命令如下。

```
ActiveCell.Formula = "=MAX( 'D:\第一季度考勤\[员工考勤.xlsx]Sheet2'!D3:D10)"
```

以上代码中，对引用的工作表要加单引号，引号内包含整个工作簿的名称及目录。



10.3 单元格的输入与输出

单元格的输入与输出包括在单元格中执行常量数值的输入与输出和公式的输入与输出，具体内容介绍如下。

10.3.1 常量的输入与输出

常量的输入与输出操作比较简单，具体内容如下。

1. 常量的输入

常量主要指数字或字符，在 VBA 代码中可以用单元格对象的 Value 属性来实现常量的输入。常量输入的命令格式如下：

```
单元格.Value=常量
```

【例 10.15】在如图 10-21 所示的工作表中，分别在 C3:C16 单元格区域中输入字符“你好”；在 F1:F10 单元格区域中输入数字 1～10。

H10 fx

	B	C	D	E	F	G	H
1					1		
2					2		
3		你好			3		
4		你好			4		
5		你好			5		
6		你好			6		
7		你好			7		
8		你好			8		
9		你好			9		
10		你好			10		
11		你好					
12		你好					
13		你好					
14		你好					
15		你好					
16		你好					
17							

图 10-21　输入字符与连续数字

在 C3：C16 单元格区域中输入字符“你好”的操作命令如下。

```
Sub 输入字符()
   Range("C3:C16").Value = "你好"
End Sub
```

【提示】Range("C3:C16").Value = "你好"可以用 Range("C3:C16") = "你好"代替。

在 F1：F10 单元格区域中输入数字 1～10 的操作命令如下。

```
Sub 输入数字()
Dim i As Integer
   For i = 1 To 10   ' 设置 1 到 10 的循环
      Cells(i, 6) = i
   Next i
End Sub
```

【提示】Cells(i,6)：在 Cells 中使用变量，单元格随着 i 的变化而变化。

2. 常量的输出

可以在程序中直接引用单元格的值参加运算或进行其他处理，具体操作内容如下。

【例 10.16】在如图 10-22 所示的工作表中，根据数量和单价计算总价，其程序代码如下。

```
Sub 单元格参加计算()
   Range("F7") = Range("D7") * Range("E7")
End Sub
```

F7 | fx | 2000

产品	单价	数量	总价
鼠标	20	100	2000

图 10-22　单元格参加计算

10.3.2 公式的输入与输出

使用 Formula 属性可以在单元格中输入公式和取得单元格公式。

1. 公式的输入

【例 10.17】在如图 10-23 所示的工作表中，根据数量和单价，在 F7 单元格中输入公式“=E7*D7”，其程序代码如下：

```
Sub 单元格中输入公式()
   Range("F7").Formula = "=E7*D7"
End Sub
```

运行该代码后，即可在 F7 单元格中自动输入公式“=E7*D7”，并根据公式计算出结果，如图 10-23 所示。

F7 | fx | =E7*D7

产品	单价	数量	总价
鼠标	20	100	2000

图 10-23　在 F7 单元格中输入公式

【例 10.18】在如图 10-24 所示的工作表中，使用代码在 E 列相应单元格中输入公式“=销售量*单价”。

	A	B	C	D	E
1	商品销售统计表				
2	品牌	产品型号	销售量	单价	总价
3	联想	G470AH-ITH	9	￥3,799	
4		Y470N-IFI(H)	12	￥5,500	
5		ThinkPad E420	20	￥4,799	
6	惠普	G4-1207tx	8	￥3,849	
7		g4-1012TX	9	￥4,099	
8		g4-1058TX	11	￥3,800	
9	苹果	Pro(MC700CH/A)	5	￥7,686	
10		Air(MC968CH/A)	12	￥7,398	
11		Air(MC965CH/A)	7	￥9,598	
12					
13					

图 10-24　在 D 列输入公式

```
Sub 在E列输入公式()
    Dim i As Integer
    Dim x As String
    Dim y As String
      For i = 3 To Range("A65536").End(xlUp).Row   ' 设置i在3到最后一行的行数
'之间循环。
        x = Cells(i,5).Offset(0, -2).Address(0, 0)
      ' 单元格偏移后的地址赋给x，Address属性返回指定单元格的地址。
        y = Cells(i, 5).Offset(0, -1).Address(0, 0)
        Cells(i, 5) = "=" & x & "*" & y
      Next i
End Sub
```

运行上述程序代码后，即可在 E 列单元格中输入相应的公式，如图 10-25 所示。

E3　=C3*D3

	A	B	C	D	E	F
1	商品销售统计表					
2	品牌	产品型号	销售量	单价	总价	
3	联想	G470AH-ITH	9	￥3,799	￥34,191	
4		Y470N-IFI(H)	12	￥5,500	￥66,000	
5		ThinkPad E420	20	￥4,799	￥95,980	
6	惠普	G4-1207tx	8	￥3,849	￥30,792	
7		g4-1012TX	9	￥4,099	￥36,891	
8		g4-1058TX	11	￥3,800	￥41,800	
9	苹果	Pro(MC700CH/A)	5	￥7,686	￥38,430	
10		Air(MC968CH/A)	12	￥7,398	￥88,776	
11		Air(MC965CH/A)	7	￥9,598	￥67,186	
12						
13						

图 10-25　填入的公式

Address 属性的代码格式为：单元格对象.Address（行是否绝对，列是否绝对）。以 A1 单元格为例，有以下几种表示方式。

$A1 表示为：

```
Range("A1").Address(False,True)
```

A $1 表示为：

```
Range("A1").Address(True, False)
```

A1 表示为：

```
Range("A1").Address(False, False)
```

【提示】Address 属性代码格式中，可将 True 用数字 1 代替，False 用 0 代替，以简化编辑。

2. 公式的输出

可以用 Formula 属性获取单元格的公式文本，并通过对话框输出表示，具体操作方法如下。

【例 10.19】使用对话框输出显示 E3 单元格的公式内容，具体操作命令如下。

```
Sub 取得单元格公式()
   MsgBox "E3    单元格的公式为：" & Range("E3").Formula
End Sub
```

运行以上代码，得到输出结果，如图 10-26 所示。



图 10-26　显示 D12 单元格公式

代码说明：Range（"E3"）.Formula 用来表示 E3 单元格中的公式。

【提示】代码中 Formula 不能省略，如果省略则显示的是值“180000”，而不是公式“=C3*D3”。



10.4 单元格删除与信息清除

可以将不需要的单元格删除，分为两种情况：一种是彻底删除单元格；另一种是只清除单元格内的信息，但保留单元格。

10.4.1 单元格删除

删除单元格后，该单元格的位置不能空着，需要执行右侧单元格左移，或下方单元格上移等操作。使用 VBA 删除单元格的具体操作内容如下。

1. 删除后右侧单元格左移

【例 10.20】删除 D4 单元格，且右侧单元格左移，具体操作命令如下。

```
Sub 删除后右侧单元格左移()
 Range("D4").Delete Shift:=xlToLeft
End Sub
```

【提示】Delete 用于删除单元格，Shift 为 Delete 方法的自变量，xlToLeft 代表单元格向左移动。

2. 删除后下方单元格上移

【例 10.21】删除 D4 单元格，且下方单元格上移，具体操作命令如下。

```
Sub 删除后下方单元格上移()
 Range("D4").Delete Shift:=xlUp
End Sub
```

3. 删除单元格所在行

【例 10.22】删除 D4 单元格所在的行，具体操作命令如下。

```
Sub 删除单元格所在行()
     Range("D4").EntireRow.Delete
End Sub
```

【提示】使用 EntireRow 属性表示指定单元格所在行。

4. 删除单元格所在列

【例 10.23】删除 D4 单元格所在的列，具体操作命令如下。

```
Sub 删除单元格所在列()
   Range("D4").EntireColumn.Delete
End Sub
```

【提示】使用 EntireColumn 属性表示指定单元格所在列。

10.4.2 单元格信息清除

单元格信息除了包含单元格的内容外，还包含单元格格式和单元格批注等。所以清除单元格信息可分为清除全部、清除格式、清除内容、清除批注和清除超链接 5 种。

1. 清除单元格全部

例如，在如图 10-27 所示的工作表中，清除 B5 单元格中的全部内容，其程序代码如下：

```
Sub 清除全部()
Range("B5").Clear
End Sub
```



图 10-27　单元格 C8 的信息

运行该程序后即可将 B5 单元格中的所有内容清除，如图 10-28 所示。

图 10-28　清除全部

2. 清除单元格格式

如果只清除单元格格式，并不清除单元格内容，其程序代码如下：

```
Sub 清除格式()
Range("B5").ClearFormats
End Sub
```

运行该程序后，即可将 B5 单元格的格式清除，如图 10-29 所示。

图 10-29　清除格式

3. 清除单元格内容

如果要清除单元格内容，保留单元格的格式及批注，可以使用下面的程序代码：

```
Sub 清除单元格内容()
Range("B5").ClearContents
End Sub
```

运行该程序后，即可将 B5 单元格的内容清除掉而保留单元格的格式及批注。



10.5 单元格的插入、隐藏及查找

当表格中的行或单元格不足时，需要插入新的行或单元格。当工作表的某些数据不需要显示时，又需要将其隐藏。利用 Excel VBA 程序代码可以轻松完成以上任务，具体内容如下。

10.5.1 单元格的插入

当表格中的单元格不足时，可以使用 Insert 方法插入相应的单元格。

1. 指定位置插入单元格

用户可以在指定位置插入同样数量的单元格。例如，要在 Sheet1 工作表“B1:C4”单元格区域的上方插入等数量的单元格，可以使用下面的程序代码：

```
Sub 在单元格上方插入()
Range("B1:C4").Insert shift:=xlDown
End Sub
```

运行该程序后，即可在“B1:C4”单元格区域上方插入高度为 3 行的单元格区域，如图 10-31 所示。

	A	B	C	D	E	F	G	H	I
1	星形			星形	星形	星形	星形		
2	星形			星形	星形	星形	星形		
3	星形			星形	星形	星形	星形		
4	星形			星形	星形	星形	星形		
5	星形	星形	星形	星形	星形	星形	星形		
6	星形	星形	星形	星形	星形	星形	星形		
7	星形	星形	星形	星形	星形	星形	星形		
8	星形	星形	星形	星形	星形	星形	星形		
9	星形	星形	星形	星形	星形	星形	星形		
10	星形	星形	星形	星形	星形	星形	星形		
11	星形	星形	星形	星形	星形	星形	星形		
12	星形	星形	星形	星形	星形	星形	星形		
13	星形	星形	星形	星形	星形	星形	星形		
14	星形	星形	星形	星形	星形	星形	星形		
15	星形	星形	星形	星形	星形	星形	星形		
16	星形	星形	星形	星形	星形	星形	星形		
17	星形	星形	星形	星形	星形	星形	星形		
18	星形	星形	星形	星形	星形	星形	星形		

图 10-31　插入单元格区域

【提示】Range("B1:C4").Insert：插入和 B1:C4 一样大小的区域。

shift:=xlDown 表示原来的单元格向下移动。Insert 方法的 shift 自变量表示原有单元格区域的移动方向。如果是 shift:=xltoright，则表示向右移动。

2. 在指定行和列前插入空行或空列

利用 VBA 实现在工作表中的指定位置插入整行或整列的代码是很容易理解的。接下来就以如图 10-32 所示的工作表为例，介绍在其中插入单行、单列、多行和多列的操作方法。

H11 星形

	A	B	C	D	E	F	G	H	I	J	K
1	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
2	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
3	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
4	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
5	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
6	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
7	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
8	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
9	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
10	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
11	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
12	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
13	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
14	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
15	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
16	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
17	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形
18	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形	星形

Sheet1 Sheet2 Sheet3

图 10-32 Sheet1 工作表

（1）在第 4 行前插入一个空行，并在第 5 行前插入一个空列，其程序代码如下：

```
Sub 指定行列前插入()
   Rows(4).Insert
   Columns(5).Insert
End Sub
```

运行该程序后，即可在工作表的第 4 行前插入一个空行，并在第 5 列前插入一个空列，如图 10-33 所示。

A1 星形

	A	B	C	D	E	F	G	H	I	J	K
1	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
2	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
3	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
4											
5	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
6	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
7	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
8	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
9	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
10	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
11	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
12	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
13	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
14	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
15	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
16	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
17	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形
18	星形	星形	星形	星形		星形	星形	星形	星形	星形	星形

Sheet1 Sheet2 Sheet3

图 10-33 插入单行和单列

（2）在第 6～9 行前插入等数量的行，并在第 G～I（7～9）列前插入等数量的列，其程序代码如下：

```
Sub 指定行列前插入多行多列()
   Rows("6:9").Insert
   Columns("G:I").Insert
End Sub
```

运行该程序后，即可在工作表的第 6～9 行前插入 4 个空行，并在第 G～I 列前插入 3 个空列，如图 10-34 所示。

	A	B	C	D	E	F	G	H	I	J
1	星形	星形	星形	星形		星形				星形
2	星形	星形	星形	星形		星形				星形
3	星形	星形	星形	星形		星形				星形
4										
5	星形	星形	星形	星形		星形				星形
6										
7										
8										
9										
10	星形	星形	星形	星形		星形				星形
11	星形	星形	星形	星形		星形				星形
12	星形	星形	星形	星形		星形				星形
13	星形	星形	星形	星形		星形				星形
14	星形	星形	星形	星形		星形				星形
15	星形	星形	星形	星形		星形				星形
16	星形	星形	星形	星形		星形				星形
17	星形	星形	星形	星形		星形				星形
18	星形	星形	星形	星形		星形				星形

图 10-34 插入多行和多列

10.5.2 单元格的隐藏

在 Excel 的基本操作中，使用功能区工具设置隐藏和取消隐藏的方法都比较简单，不再介绍。利用 VBA 中单元格的 Hidden 属性可实现隐藏或取消隐藏的功能，具体操作方法如下。

例如，隐藏第 3～6 行和第 2～5 列，其程序代码如下：

```
Sub 隐藏行和列()
  Rows("3:6").Hidden = True
  Columns("B:E").Hidden = True
End Sub
```

运行该语句后，即可在工作表中隐藏第 3～6 行和第 2～5 列单元格，如图 10-35 所示。

	A	F	G	H	I
1	星形	星形	星形	星形	星形
2	星形	星形	星形	星形	星形
7	星形	星形	星形	星形	星形
8	星形	星形	星形	星形	星形
9	星形	星形	星形	星形	星形
10	星形	星形	星形	星形	星形
11	星形	星形	星形	星形	星形
12	星形	星形	星形	星形	星形
13	星形	星形	星形	星形	星形
14	星形	星形	星形	星形	星形

图 10-35 隐藏整行和整列

若要取消隐藏第 3～6 行和第 2～5 列，可以使用下面的程序代码：

```
Sub 取消隐藏行和列()
  Rows("3:6").Hidden = False
  Columns("B:E").Hidden = False
End Sub
```

运行该语句后，即可取消工作表中隐藏的第 3～6 行和第 2～5 列单元格，如图 10-36 所示。

	A	B	C	D	E	F
1	星形	星形	星形	星形	星形	星形
2	星形	星形	星形	星形	星形	星形
3	星形	星形	星形	星形	星形	星形
4	星形	星形	星形	星形	星形	星形
5	星形	星形	星形	星形	星形	星形
6	星形	星形	星形	星形	星形	星形
7	星形	星形	星形	星形	星形	星形
8	星形	星形	星形	星形	星形	星形
9	星形	星形	星形	星形	星形	星形
10	星形	星形	星形	星形	星形	星形
11	星形	星形	星形	星形	星形	星形
12	星形	星形	星形	星形	星形	星形

Sheet1 Sheet2 Sheet3

图 10-36　取消隐藏整行和整列

【提示】单元格对象的 Hidden 方法可以指定单元格区域的显示属性，当值为 True 时隐藏，当值为 False 时取消隐藏。

在 Excel 中无法隐藏单个的单元格，只能通过隐藏指定单元格所在行或列来实现隐藏该单元格的目的。

例如，隐藏 B4 所在的行和列，其程序代码如下：

```
Sub 隐藏单个单元格所在的行和列()
 Range("B4").EntireRow.Hidden = True
 Range("B4").EntireColumn.Hidden = True
End Sub
```

运行该程序后，即可隐藏 B4 所在的行和列，如图 10-37 所示。

	A	C	D	E	F	G
1	星形	星形	星形	星形	星形	星形
2	星形	星形	星形	星形	星形	星形
3	星形	星形	星形	星形	星形	星形
5	星形	星形	星形	星形	星形	星形
6	星形	星形	星形	星形	星形	星形
7	星形	星形	星形	星形	星形	星形
8	星形	星形	星形	星形	星形	星形
9	星形	星形	星形	星形	星形	星形
10	星形	星形	星形	星形	星形	星形
11	星形	星形	星形	星形	星形	星形
12	星形	星形	星形	星形	星形	星形
13	星形	星形	星形	星形	星形	星形

Sheet1 Sheet2 Sheet3

图 10-37　隐藏单个单元格所在的行和列

也可以隐藏单元格区域所在的行和列，例如隐藏 B4:D6 所在区域的行和列，其程序代码如下：

```
Sub 隐藏单元格区域所在的行和列()
 Range("B4:D6").EntireRow.Hidden = True
 Range("B4:D6").EntireColumn.Hidden = True
End Sub
```

运行该程序后，即可隐藏工作表中的 B4:D6 区域的行和列，如图 10-38 所示。

	A	E	F	G	H	I
1	星形	星形	星形	星形	星形	星形
2	星形	星形	星形	星形	星形	星形
3	星形	星形	星形	星形	星形	星形
7	星形	星形	星形	星形	星形	星形
8	星形	星形	星形	星形	星形	星形
9	星形	星形	星形	星形	星形	星形
10	星形	星形	星形	星形	星形	星形
11	星形	星形	星形	星形	星形	星形
12	星形	星形	星形	星形	星形	星形
13	星形	星形	星形	星形	星形	星形
14	星形	星形	星形	星形	星形	星形
15	星形	星形	星形	星形	星形	星形

Sheet1 Sheet2 Sheet3

图 10-38　隐藏单元格区域所在的行和列

利用 VBA 隐藏单元格的方法很灵活，用户可以根据自己的实际需要或使用习惯，选择合适的方法。

10.5.3 单元格的查找

在编写代码时，常需要依据指定内容在指定区域查找符合条件的单元格，这时就需要用查找的方法。使用 VBA 程序代码查找单元格的操作方法如下。

1. 使用 Find 方法进行查找

在 Excel 中可以通过【查找和替换】工具找到需要的单元格数据，同样也可以使用 VBA 程序代码实现。这里主要介绍 Find 方法查找，具体内容介绍如下。

【例 10.24】在如图 10-39 所示的工作表中，查询 G3 单元格的产品型号所在的行数，并将查找结果表示在 G4 单元格中，具体操作命令如下。

```
Sub 查找姓名所在行()
 Range("G4") = Range("B:B").Find(Range("G3")).Row
End Sub
```

	A	B	C	D	E	F	G
1	商品销售统计表						
2	品牌	产品型号	销售量	单价	总价		
3	联想	G470AH-ITH	9	￥3,799	￥34,191		G4-1207tx
4		Y470N-IFI(H)	12	￥5,500	￥66,000		
5		ThinkPad E420	20	￥4,799	￥95,980		
6	惠普	G4-1207tx	8	￥3,849	￥30,792		
7		g4-1012TX	9	￥4,099	￥36,891		
8		g4-1058TX	11	￥3,800	￥41,800		
9	苹果	Pro(MC700CH/A)	5	￥7,686	￥38,430		
10		Air(MC968CH/A)	12	￥7,398	￥88,776		
11		Air(MC965CH/A)	7	￥9,598	￥67,186		
12							
13							

图 10-39　按产品型号查找

【提示】 Range("B:B").Find 表示在 B 列进行查找；Range("G3")把 G3 的值作为要查找的内容。

运行该程序后，即可在 G4 单元格中显示出查找后的结果。

2. 使用工作表函数进行查找

使用工作表函数同样可以实现单元格查找功能。这里主要介绍两个函数：Match 函数和 Vlookup 函数。其中 Match 函数可以返回单元格的位置，而 Vlookup 函数可以返回单元格的内容。

（1）使用 Match 函数返回所查找的单元格位置。

在如图 10-40 所示的工作表中，查询 G3 单元格的产品型号所在的行数，并将查找结果表示在 G4 单元格中，使用 Match 函数的操作命令如下。

```
Sub 使用工作表函数查找姓名所在行()
 Range("G4") = Application.Match(Range("G3"), Range("B:B"), 0)
End Sub
```

（2）使用 Vlookup 函数返回查找的内容。

例如，在如图 10-40 所示的工作表中，要根据 G6 单元格内容，在 B 列查找并返回该型号所对应的单价，并把查找的结果填入到 G7 单元格中，使用 Find 实现的具体操作命令如下。

```
Sub 查找型号单价()
 Dim Mrow As Integer
 Mrow = Range("B:B").Find(Range("G6")).Row  ' 在 B 列中查找匹配 G6 单元格的行
 Range("G7") = Range("D" & Mrow)  ' 在 G7 单元格中返回对应行 D 列的数据信息
End Sub
```

	B	C	D	E	F	G
1	商品销售统计表					
2	产品型号	销售量	单价	总价		
3	G470AH-ITH	9	￥3,799	￥34,191		G4-1207tx
4	Y470N-IFI(H)	12	￥5,500	￥66,000		6
5	ThinkPad E420	20	￥4,799	￥95,980		
6	G4-1207tx	8	￥3,849	￥30,792		Pro(MC700CH/A)
7	g4-1012TX	9	￥4,099	￥36,891		
8	g4-1058TX	11	￥3,800	￥41,800		
9	Pro(MC700CH/A)	5	￥7,686	￥38,430		
10	Air(MC968CH/A)	12	￥7,398	￥88,776		
11	Air(MC965CH/A)	7	￥9,598	￥67,186		
12						
13						

图 10-40　查找单价

使用工作表函数 Vlookup 进行查找，操作命令会更加简洁，具体命令如下。

```
Sub 使用函数查找型号单价()
 Range("G7") = Application.VLookup(Range("G6"), Range("B:E"), 3, 0)
End Sub
```

【提示】 上述命令中，“B:E”表示匹配查找的数据区域 B 列到 E 列。“3”表示“B:E”区域的第三列所在的单元格的信息为要返回的数据。



10.6 合并和设置单元格

除了以上介绍的单元格的操作内容外，还有其他单元格的操作内容。本节将详细介绍单元格的格式设定和合并单元格等操作内容。

10.6.1 合并单元格

一个复杂完善的工作表，并不是所有单元格都一样大小的。例如整个表格的标题，一般在表格的上方居中，这样的话标题就不可能只占一个单元格。在 Excel 基本操作中，可以通过合并单元格的方式解决以上问题。

利用 VBA 代码可以实现单元格的合并操作，同时在取消合并时，每个单元格中都会保留原合并单元格的内容。

1. 合并单元格时连接各单元格的文本

如果合并单元格时，原来的多个单元格中都有数据，可以通过命令将这些单元格的数据连接起来，并显示在合并后的单元格中。

【例 10.25】将如图 10-41 所示的工作表中的 B5 和 C5 单元格进行合并，并将各单元格内容连接起来显示在合并后的单元格区域中，具体操作命令如下。

```
Sub 合并单元格()
Dim x As String  ' 定义字符串变量 x，用于显示合并后的文本
Dim y As Range  ' 定义 Range 类型变量 y
   For Each y In Range("B5:C5")  ' 设置变量 y 的循环区域
     x = x & y.Value  ' 将单元格区域中的内容连接起来保存在字符串变量 x 中
   Next
     Application.DisplayAlerts = False ' 禁止在合并单元格时显示警告功能
     Range("B5:C5").Merge  ' 合并单元格区域“B5:C5”
     Range("B5:C5").Value = x  ' 将 x 的值赋给单元格区域“B5:C5”
     Application.DisplayAlerts = True  ' 恢复应用程序显示警告和消息功能
End Sub
```

运行该程序，得到如图 10-42 所示的结果。

	A	B	C
1			
2			
3			
4			
5		ABC	DEF
6			

图 10-41　合并前的单元格

	A	B	C
1			
2			
3			
4			
5		ABCDEF	
6			

图 10-42　合并后的结果

在上述程序中指定了固定的单元格区域进行合并，在实际应用中可能会对任意单元格区域进行合并，所以可以通过 VBA 设定合并单元格的操作代码，当选择好单元格区域后，直接执行该程序代码即可。合并任意单元格区域的操作命令如下。

```
Sub 合并任意所选区域()
   Dim x As String
   Dim y As Range
   If TypeName(Selection) = "Range" Then
      For Each y In Selection
         x = x & y.Value
      Next
         Application.DisplayAlerts = False
         Selection.Merge
         Selection.Value = x
         Application.DisplayAlerts = True
   End If
End Sub
```

2. 取消合并单元格时在每个单元格中保留内容

有单元格合并就有取消合并，具体操作内容如下。

【例 10.26】取消如图 10-43 所示的 A 列中的合并单元格，并且将数据分别保留在取消合并后的所有单元格中，具体操作命令如下。

```
Sub 取消合并()
Dim i As Integer
Dim x As String
Dim y As Integer
  For i = 1 To 100         ' 从第 1 行开始，到第 100 行
     x = Cells(i,1).Value   ' 将 A 列每个合并单元格中的内容赋值给 x
     y = Cells(i,1).MergeArea.Count  ' 取得合并区域中的单元格数量
     Cells(i, 1).UnMerge '利用 UnMerge 方法取消合并单元格
     Range(Cells(i, 1), Cells(i + y - 1, 1)).Value = x
     ' 将原合并单元格中的内容赋值给取消合并单元格后的区域
     i = i + y - 1 '调整循环变量 i 的值，使下一次循环从该合并区域的下一个单元格开始
  Next i
End Sub
```

运行该程序，取消 A 列中合并的单元格，并且取消合并的单元格中都保留了原单元格的数据内容，如图 10-44 所示。

图 10-43　合并的单元格

图 10-44　合并单元格



10.6.2 设置单元格格式

为了使表格更加美观，使单元格内容信息能更加清晰地表示，需要对单元格进行格式化操作。可以通过右击单元格，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，打开【设置单元格格式】对话框，从中对单元格格式进行调整。同时也可以使用 VBA 程序代码完成类似操作。

1. 设置字体等格式

单元格格式最基本的就是字体样式，在 VBA 中可以使用 Font 对象的 Name、Size 等属性设置字体的格式。

【例 10.27】对如图 10-45 所示的工作表中的表格设置字体格式，其中标题的字体设置为“蓝色”、“18 号”、“华文隶书”，标题底纹为“紫色”；第二行选项的字体设置为“黑色”、“14 号”、“华文中宋”；所有表格数据均设置为居中对齐。

	A	B	C	D	E	F
1			商品销售统计表			
2	品牌	产品型号	销售量	单价	总价	
3	联想	G470AH-IT	9	3799	34191	
4		Y470N-IFI	12	5500	66000	
5		ThinkPad	20	4799	95980	
6		g4-1012TX	9	4099	36891	
7		g4-1058TX	11	3800	41800	
8		Air(MC968	12	7398	88776	
9		Air(MC965	7	9598	67186	
10						
11						

图 10-45　要设置格式的工作表

完成以上内容的具体操作命令如下。

```
Sub 设置字体格式()
    Range("A1").Select
    With Selection
        .Font.Name = "华文隶书"
        .Font.Size = 18
        .Font.ColorIndex = 5
        .Interior.ColorIndex = 39
    End With
    Range("A2:E2").Select
    With Selection.Font
        .Name = "华文中宋"
        .Size = 14
    End With
    Range("A3:A9,B3:B9,C3:C9,D3:D9,E3:E9").Select
    With Selection.Font
        .Name = "Times New Roman"
        .Size = 11
    End With
    Range("A2:E9").Select
```

```
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
    End With
    Range("A1").Select
End Sub
```

运行该程序，得到设置后的效果，如图 10-46 所示。

商品销售统计表				
品牌	产品型号	销售量	单价	总价
联想	G470AH-ITH	9	3799	34191
	Y470N-IFI(H)	12	5500	66000
	ThinkPad E420	20	4799	95980
	g4-1012TX	9	4099	36891
	g4-1058TX	11	3800	41800
	Air(MC968CH/A)	12	7398	88776
	Air(MC965CH/A)	7	9598	67186

图 10-46　设置格式后的效果

2. 为表格添加边框

可以使用 Range 对象的 Borders 集合快速地对单元格区域的每个边框应用相同的格式，并且 Range 对象的 BorderAround 方法可以为单元格区域添加一个外边框。

【例 10.28】为如图 10-47 所示的“商品销售统计表”工作表添加边框，具体操作命令如下。

```
Sub 添加边框()
Dim i As Range
 Set i = Range("A1:E9")  ' 设置对象变量为单元格区域“A1:E9”
    With i.Borders(xlInsideVertical)
        .LineStyle = xlContinuous  ' 设置边框的线条样式
        .Weight = xlThin  ' 设置边框线条的粗细
        .ColorIndex = 55  ' 设置边框的颜色
    End With
    With i.Borders(xlInsideHorizontal)
        .LineStyle = xlDash
        .Weight = xlThin
        .ColorIndex = 55
    End With
    i.BorderAround xlContinuous, xlMedium, 55
 Set i = Nothing    ' 从内存中释放对象变量，与 Set 相对应
End Sub
```

运行该程序，为“商品销售统计表”添加边框，如图 10-47 所示。

	A	B	C	D	E
1	商品销售统计表				
2	品牌	产品型号	销售量	单价	总价
3	联想	G470AH-ITH	9	3799	34191
4		Y470N-IFI(H)	12	5500	66000
5		ThinkPad E420	20	4799	95980
6		g4-1012TX	9	4099	36891
7		g4-1058TX	11	3800	41800
8		Air(MC968CH/A)	12	7398	88776
9		Air(MC965CH/A)	7	9598	67186
10					
11					

图 10-47　为表格添加边框

【提示】上述程序代码中的 Borders(index)属性可以返回单个 Border 对象，其中参数的取值及意义如表 10-2 所示。

表 10-2　index 的取值及意义

常量	值	说明
xlDiagonalDown	5	斜下边框
xlDiagonalUp	6	斜下边框
xlEdgeLeft	7	左边框
xlEdgeTop	8	顶部边框
xlEdgeBottom	9	底部边框
xlEdgeRight	10	右边框
xlInsideVertical	11	内部垂直
xlInsideHorizontal	12	内部水平

利用 Range 对象的 BorderAround 方法可以为单元格区域添加整个区域的外边框，其语法格式为：

```
BorderAround(LineStyle,Weight,ColorIndex,Color)
```

【提示】ColorIndex 参数和 Color 参数都可以指定边框颜色，但是 ColorIndex 参数可以指定当前调色板中的颜色，而 Color 参数以 RGB 值指定边框的颜色，可以是当前调色板以外的颜色。但是以上两种参数不能同时使用。



10.7　综合实战 1——为销售清单实现公式快速自动填充

在 Excel 操作中，会遇到同一列执行相同计算公式的现象，这时可以利用 VBA 代码实现公式的自动填充。具体操作内容介绍如下。

例如，在如图 10-48 所示的“笔记本销售清单”工作表中，要求当单元格 C、D 和 E 列中的单元格内容发生改变后，在 G 列中会自动填充上用于计算“实际金额”的公式。其中“实际金额=数量*单价*折扣率”，具体操作命令如下。

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
    Dim Trow As Integer
    Dim Bcol As String, Ccol As String
    Dim Dcol As String, Ecol As String
       Trow = Target.Row  '将当前编辑的单元格的行数赋值给变量 Trow
     If Trow > 2 And Len(Range("C" & Trow)) > 0 And Len(Range("D" & Trow)) > 0 And
Len(Range("E" & Trow)) > 0 Then
    '从第 3 行开始，只有当“数量”、“单价”和“折扣率”项都填入内容后，才执行下一条语句
       Ccol = Range("C" & Trow).Address(0, 0)
       Dcol = Range("D" & Trow).Address(0, 0)
       Ecol = Range("E" & Trow).Address(0, 0)
       Range("G" & Trow).Formula = "=" & Ccol & "*" & Dcol & "*" & Ecol
     End If
    End Sub
```

【提示】Worksheet_Change 表示当工作表中单元格发生修改时运行程序。

此时，当改变单元格内容时，例如将 C3 单元格中的“100”改变“300”后，即可在对应的 G3 单元格中计算出该笔记本的“实际金额”，如图 10-48 所示。

	A	B	C	D	E	F	G
1	笔记本销售清单						
2	时间	产品型号	数量	单价	折扣率	折扣额	实际金额
3	2012-1-10	G470AH					
4	2012-1-10	Y470N-IFI(H)					
5	2012-1-10	ThinkPad E420					
6	2012-1-11	G470AH					
7	2012-1-11	Y470N-IFI(H)					
8	2012-1-11	ThinkPad E420					
9	2012-1-12	G470AH					
10	2012-1-12	Y470N-IFI(H)					
11	2012-1-12	ThinkPad E420					
12	2012-1-13	G470AH					
13	2012-1-13	Y470N-IFI(H)					
14	2012-1-13	ThinkPad E420					
15	2012-1-14	G470AH					
16	2012-1-14	Y470N-IFI(H)					
17	2012-1-14	ThinkPad E420					
18	2012-1-15	G470AH					
19	2012-1-15	Y470N-IFI(H)					
20	2012-1-15	ThinkPad E420					

图 10-48　笔记本销售清单

当输入笔记本销售信息后，系统便会自动计算出对应的“实际金额”，其中“折扣额”列的数据也可以使用上述方式添加运算公式，添加后的运算结果如图 10-49 所示。

	B	C	D	E	F	G
1	笔记本销售清单					
2	产品型号	数量	单价	折扣率	折扣额	实际金额
3	G470AH	3	￥3,799	0.9	￥1,139.70	￥10,257.30
4	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00
5	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30
6	G470AH	7	￥3,849	0.85	￥4,041.45	￥22,901.55
7	Y470N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00
8	ThinkPad E420	13	￥4,600	0.9	￥5,980.00	￥53,820.00
9	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20
10	Y470N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00
11	ThinkPad E420	6	￥4,799	0.85	￥4,319.10	￥24,474.90
12	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25
13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80	￥18,696.60
14	ThinkPad E420	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	Y470N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	ThinkPad E420	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	Y470N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20
20	ThinkPad E420	14	￥4,600	0.75	￥16,100.00	￥48,300.00

图 10-49　执行公式自动填充





10.8 综合实战 2——为销售记录自动填充颜色

在实际应用中，可以利用 VBA 代码将不同信息的内容用不同的颜色进行区分，具体操作方法如下。

为如图 10-50 所示的“笔记本销售清单”工作表中不同产品的销售记录使用不同的颜色进行表示。

	B	C	D	E	F	G	H	I	J
1	笔记本销售清单								
2	产品型号	数量	单价	折扣率	折扣额	实际金额			
3	G470AH	3	￥3,799	0.9	￥1,139.70	￥10,257.30		G470AH	
4	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00		Y470N-IFI(H)	
5	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30		ThinkPad E420	
6	G470AH	7	￥3,849	0.85	￥4,041.45	￥22,901.55			
7	Y470N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00			
8	ThinkPad E420	13	￥4,600	0.9	￥5,980.00	￥53,820.00			
9	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20			
10	Y470N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00			
11	ThinkPad E420	6	￥4,799	0.85	￥4,319.10	￥24,474.90			
12	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25			
13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80	￥18,696.60			
14	ThinkPad E420	12	￥4,600	0.75	￥13,800.00	￥41,400.00			
15	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25			
16	Y470N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00			
17	ThinkPad E420	6	￥4,799	0.8	￥5,758.80	￥23,035.20			
18	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85			
19	Y470N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20			
20	ThinkPad E420	14	￥4,600	0.75	￥16,100.00	￥48,300.00			

图 10-50 为销售记录的自动填充颜色

具体操作步骤如下。

01 在 VBA 代码窗口中，输入以下程序代码。

```
    Private Sub Worksheet_Change(ByVal Target As Range)
    On Error GoTo AAA
     Dim Trow As Integer
     Dim Frow As Integer
     Dim ys As Integer
        Trow = Target.Row    ' 将当前的行号赋值给变量 Trow
     If Trow > 2 Then     ' 如果当前行号大于 2，则执行下一条语句，否则退出事件程序
        If Target.Column = 2 Then
    ' 判断当前列数是否为 2（因为颜色和部门有关，所以只让事件代码作用于产品型号所在的第 2 列）。
'如果不为 2（即改变其他列中的单元格时），则退出事件程序
           If Len(Target) > 0 Then  ' 如果当前单元格的内容不为空，则执行下一条语句
             Frow = Range("I:I").Find(Target).Row
    ' 根据当前产品型号在 I 列中进行查找，并将查找到的单元格的行号赋值给变量 Frow
             ys = Range("J" & Frow).Interior.ColorIndex  '取得 J 列相对应产品型号单元
格'的颜色
             Range(Cells(Trow, 1), Cells(Trow, 7)).Interior.ColorIndex = ys
     ' 将当前单元格所在行的第 1 列到第 7 列的单元格区域的填充颜色值设置为 ys
           Else
```

```
            Range(Cells(Trow, 1), Cells(Trow, 7)).Interior.ColorIndex = 0
    '如果不符合条件，则将当前单元格所在行的第 1 列到第 7 列的单元格区域的填充颜色值设置为 0，即
'无颜色填充
          End If
        End If
     End If
     Exit Sub
    AAA:
       Range(Cells(Trow, 1), Cells(Trow,7)). _
       Interior.ColorIndex = 0
    End Sub
```

02 在代码窗口中新插入一个模块，输入以下程序代码。

```
    Sub 刷新填充颜色()
    Dim i As Integer
      For i = 3 To Range("A65536").End(xlUp).Row
        Cells(i, 2) = Cells(i, 2)
      Next i
    End Sub
```

03 运行“刷新填充颜色”过程，即可将工作表中的数据填充上设置的各部门的对应颜色，如图 10-51 所示。

	B	C	D	E	F	G	H	I	J
1	笔记本销售清单								
2	产品型号	数量	单价	折扣率	折扣额	实际金额			
3	G470AH	3	￥3,799	0.9	￥1,139.70	￥10,257.30		G470AH	
4	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00		Y470N-IFI(H)	
5	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30		ThinkPad E420	
6	G470AH	7	￥3,849	0.85	￥4,041.45	￥22,901.55			
7	Y470N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00			
8	ThinkPad E420	13	￥4,600	0.9	￥5,980.00	￥53,820.00			
9	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20			
10	Y470N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00			
11	ThinkPad E420	6	￥4,799	0.85	￥4,319.10	￥24,474.90			
12	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25			
13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80	￥18,696.60			
14	ThinkPad E420	12	￥4,600	0.75	￥13,800.00	￥41,400.00			
15	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25			
16	Y470N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00			
17	ThinkPad E420	6	￥4,799	0.8	￥5,758.80	￥23,035.20			
18	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85			
19	Y470N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20			
20	ThinkPad E420	14	￥4,600	0.75	￥16,100.00	￥48,300.00			

图 10-51　为各部门填充颜色



10.9　综合实战 3——统计单元格区域的数目

在 Excel 中会遇到统计一定区域中单元格数量的要求，可以使用以下程序代码实现该功能，具体操作命令如下。

```
    Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```




```
    MsgBox "您好，选中区域的单元格个数为：" & Selection.Count
End Sub
```

代码输入完毕后，返回工作表中选中任意单元格区域，例如这里选择“B4：F10”单元格区域，随即便会弹出显示所选区域单元格数目的提示框，如图 10-52 所示。

B4　Y470N-IFI(H)

	A	B	C	D	E	F
1	笔记本销售清单					
2	时间	产品型号	数量	单价	折扣率	折扣额
3	2012-1-10	G470AH	3	￥3,799	0.9	￥1,139.70
4	2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00
5	2012-1-10	ThinkPad E420				439.70
6	2012-1-11	G470AH				041.45
7	2012-1-11	Y470N-IFI(H)				998.00
8	2012-1-11	ThinkPad E420				980.00
9	2012-1-12	G470AH				357.80
10	2012-1-12	Y470N-IFI(H)				375.00
11	2012-1-12	ThinkPad E420	6	￥4,799	0.85	￥4,319.10
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75
13	2012-1-13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80

Microsoft Excel
您好，选中区域的单元格个数为：35
确定

Sheet1　Sheet2　Sheet3

图 10-52　显示选中单元格数目

除此之外，还可以统计已指定名称的单元格区域的单元格数，具体操作命令如下。

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    X = Range("单价记录").Count
End Sub
```

【提示】可以右击选中的单元格区域，在弹出的快捷菜单中选择【定义名称】菜单命令，为单元格区域指定名称。

如果需要统计所选区域的行数，可以使用以下程序命令。

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    MsgBox "您好，选中区域的行数为：" & Selection.Rows.Count
End Sub
```

当选择某行时，系统即可弹出选择区域的行数信息，如图 10-53 所示。

A14　2012-1-13

	A	B	C	D	E	F
10	2012-1-12	Y470N-IFI(H)	5	￥5,500	0.95	￥1,375.00
11	2012-1-12	ThinkPad E420	6	￥4,799	0.85	￥4,319.10
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75
13	2012-1-13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80
14	2012-1-13	ThinkPad E420				￥13,800.00
15	2012-1-14	G470AH				￥8,547.75
16	2012-1-14	Y470N-IFI(H)				￥7,150.00
17	2012-1-14	ThinkPad E420				￥5,758.80
18	2012-1-15	G470AH				￥1,347.15
19	2012-1-15	Y470N-IFI(H)				￥6,598.80
20	2012-1-15	ThinkPad E420	14	￥4,600	0.75	￥16,100.00
21						
22						
23						

Microsoft Excel
您好，选中区域的行数为：3
确定

Sheet1　Sheet2　Sheet3

图 10-53　显示选择区域的行数

如果需要统计所选区域的列数，可以使用以下程序命令。

```
Private Sub Worksheet SelectionChange(ByVal Target As Range)
   MsgBox "您好，选中区域的列数为：" & Selection.Columns.Count
End Sub
```

当选择某列时，系统即可弹出选择区域的列数信息，如图 10-54 所示。

B6　G470AH

	A	B	C	D	E	F
1	笔记本销售清单					
2	时间	产品型号	数量	单价	折扣率	折扣额
3	2012-1-10	G470AH	3	￥3,799	0.9	￥1,139.70
4	2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00
5	2012-1-10	ThinkPad E420	6	￥4,799	0.95	￥1,439.70
6	2012-1-11	G470AH	7	￥3,849	0.85	￥4,041.45
7	2012-1-11	Y470N-IFI(H)	8			3.00
8	2012-1-11	ThinkPad E420	13			0.00
9	2012-1-12	G470AH	11			7.80
10	2012-1-12	Y470N-IFI(H)	5			5.00
11	2012-1-12	ThinkPad E420	6			9.10
12	2012-1-13	G470AH	7			5.75
13	2012-1-13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80

Sheet1　Sheet2　Sheet3

Microsoft Excel

您好，选中区域的列数为：3

确定

图 10-54　显示选择区域的列数



10.10　综合实战 4——确定单元格区域的数据类型

可以通过 Excel 内置函数和 VBA 中包含的函数确定单元格中的数据类型，同时也可以通过 Function 自定义一个可以判断数据类型的函数，然后通过调用该函数来确定单元格中数据的类型，具体操作内容如下。

首先需要创建一个可以判断数据类型的自定义函数，操作命令如下。

```
Function CellType(Rng)   ' 创建自定义函数 CellType
Dim Cella As Range
Set Cella = Rng.Range("A1")
Select Case True
    Case IsEmpty(Cella)
        CellType = "Blank"
    Case Application.IsText(Cella)
        CellType = "Text"
    Case Application.IsLogical(Cella)
        CellType = "Logical"
    Case Application.IsErr(Cella)
        CellType = "Error"
    Case IsDate(Cella)
        CellType = "Date"
    Case InStr(1, Cella.Text, ":") <> 0
        CellType = "Time"
    Case InStr(1, Cella.Text, "￥") <> 0
        CellType = "Money"
```




```
        Case IsNumeric(Cella)
            CellType = "Number"
    End Select
    End Function
```

上述自定义函数创建之后，需要在该模块中输入调用指令，具体操作命令如下。

```
    Sub 类型()
    Dim a As Variant
      a = CellType(Selection)
      MsgBox "您好，选择的类型为：" & a, vbInformation
    End Sub
```

代码输入完毕后，返回工作表，为了方便操作，可以为该程序创建执行按钮。当选中某单元格后，单击创建的执行按钮，系统便会弹出所选单元格中数据的类型，如图 10-55 所示。



图 10-55　显示单元格数据类型

【提示】程序执行按钮的添加可以参考第 11 章中窗体控件的相关操作内容。

如果选中的是单元格区域，则在执行数据类型判断时，只能返回该区域左上角单元格内数据的数据类型，如图 10-56 所示。

	A	B	C	D	E	F
13	2012-1-13	Y470N-IFI(H)	4	￥5,499	0.85	￥6,598.80
14	2012-1-13	ThinkPad E420	12	￥4,600	0.75	￥13,800.00
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75
16	2012-1-14	Y470N-IFI(H)	13	￥5,500	0.9	￥7,150.00
17	2012-1-14	ThinkPad E420	6	￥4,799	0.8	￥5,758.80
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15
19	2012-1-15	Y470N-IFI(H)	8			
20	2012-1-15	ThinkPad E420	14			

D15　3799

Microsoft Excel
您好，选择的类型为：Money
确定

图 10-56　显示区域左上角单元格的数据类型



10.11 高手私房菜

技巧 1：快速清除单元格批注

如果要清除单元格的批注，而对内容和格式不影响，则可以使用下面的程序代码：

```
Sub 清除批注()
Range("B5").ClearComments
End Sub
```

运行该程序后，即可清除 B5 单元格的批注，而不影响单元格内容和格式。

技巧 2：快速查找重复的数据

当工作表数据中要查询的数据有多个重复时，可以使用循环查询的方式来完成操作。例如，在笔记本销售的工作表中，找出所有 G470AH-ITH 笔记本的销售记录，并将最后一次的订购量值填入 H3 单元格中，具体操作命令如下。

```
Sub 查找3月份订购量()
Dim i As Integer    ' 定义i为整数，程序中i是C列中G470AH-ITH笔记本的记录数
Dim Mrg As Range    ' 定义Mrg为单元格对象，Mrg作为查找的开始单元格
   i = Application.CountIf(Range("C:C"), Range("H2"))
      '调用工作表函数Count计算C列中H2单元格中产品名的个数
  Set Mrg = Range("C3")
          ' 设置Mrg的初始单元格为C3，即在下面的查找过程中首先从C3开始查找
   For s = 1 To i  ' 要根据G470AH-ITH的个数（i值）设置查询的次数
      Set Mrg = Range("C:C").Find(Range("H3"), Mrg)
      ' 每一次新的查找都从查找到的单元格开始
   Next s
  Range("H3") = Mrg.Offset(0, 1)
          ' 单元格H3的值等于查找到的单元格右边一个单元格的值，即入库价格
End Sub
```

运行该程序，可在 H3 单元格中查找出 G470AH-ITH 笔记本最后一个月的订购量。

技巧 3：轻松查找区域中的最大值

在实际应用中经常会遇到要查找工作表中最大值的现象，如查找考试成绩最高的学生。

使用 VBA 可以找出区域中的最大值，而且还可以为最大值字体设置颜色等格式，具体操作命令如下。

```
Sub 突出显示最大值()
Dim WorkRange As Range
Dim MaxVal As Double
```

```
Dim Num As Variant
  If TypeName(Selection) <> "Range" Then  ' 判断所选区域是否为 Range 对象
     Exit Sub
  End If
  If Selection.Count = 1 Then   ' 如果所选区域的单元格个数为 1，则执行下一条语句
    Set WorkRange = Cells  ' 当选中单个单元格时，系统会自动选择整个工作表
  Else
     Set WorkRange = Selection
  End If
  MaxVal = Application.Max(WorkRange)  '获取该选区的最大值
  For Each Num In WorkRange
     If Num.Value = MaxVal Then
        Num.Font.ColorIndex = 3  ' 将最大值字体颜色设置为红色
     End If
  Next Num
End Sub
```

返回工作表，选中要查找的单元格区域，然后运行该程序，即可将所选区域中的最大值以红色字体显示出来。

第 11 章　控制工具与窗体控件的相关操作

一个完整的应用必不可少地需要一些控件与窗体来实现其功能。控件工具可以帮助用户高效率地完成很多任务，从而达到事半功倍的效果。本节将详细介绍控制工具与窗体控件的操作内容。



11.1 对表格内容进行处理

在使用 Excel 2010 进行数据处理与分析工作时，数据信息所在的单元格样式往往不能满足用户的需要，这时就有必要对单元格的样式进行设计。并且当工作表中存有大量数据时，为便于查看，可以使用冻结窗格的功能。

11.1.1 实现英文名称标准化

在表格中录入英文文本时，需要注意其格式。标准的英文录入应该是首字母大写，而其他字母全部为小写。所以在表格编辑时，需要将一些书写不规范的英文名称转化为标准形式，这样会更便于信息的阅读和处理。

【例 11.1】在“员工考核报表”中添加控件按钮，实现英文名称标准化显示，具体操作步骤如下。

01 在如图 11-1 所示的“员工考核报表”工作表中，选择【开发工具】选项卡，在【控件】选项区中单击【插入】按钮，即可打开【表单控件】和【ActiveX 控件】。



图 11-1 “员工考核报表”工作表

02 从中单击【命令按钮】按钮，在工作表的适当位置待鼠标指针变为“+”形状时，按住鼠标左键不放，拖至适当大小后释放鼠标左键，即可添加一个名为“CommandButton1”的命令按钮，如图 11-2 所示。

员工考核报表						
编号	姓名	所属部门	务技能(3(	勤情况(1(	织能力(1(	综合成绩
1001	lifang	销售部	25	9	9	91
1002	duanling	财务部	26	10	8	88
1003	chenyan	销售部	24	7	8	80.5
1004	sunshi	销售部	29	8	7	83
1005	zhengjing	财务部	23	9	7	79
1006	yangp	销售部	25	10	8	82
1007	wangyi	销售部	24	10	6	81
1008	liwan	财务部	28	9	6	84
1009	wansi	行政部	20	8	6	71
1010	wanse	后勤部	21	10	6	75
1011	chensan	财务部	26	9	7	83
1012	zanlen	财务部	25	8	7	79
1013	xubin	销售部	28	10	8	89.5
1014	guolou	人事部	27	8	9	87
1015	sinna	销售部	24	9	8	80.5
1016	wanyin	人事部	26	10	7	83

CommandButton1

(Ctrl)

图 11-2 添加一个命令按钮

03 右击添加的命令按钮，在弹出的快捷菜单中选中【属性】菜单命令，弹出【属性】对话框，选择【按字母序】选项卡，然后在【Caption】文本框中输入“英文名称标准化”，即可修改添加的命令按钮的名称，如图 11-3 所示。

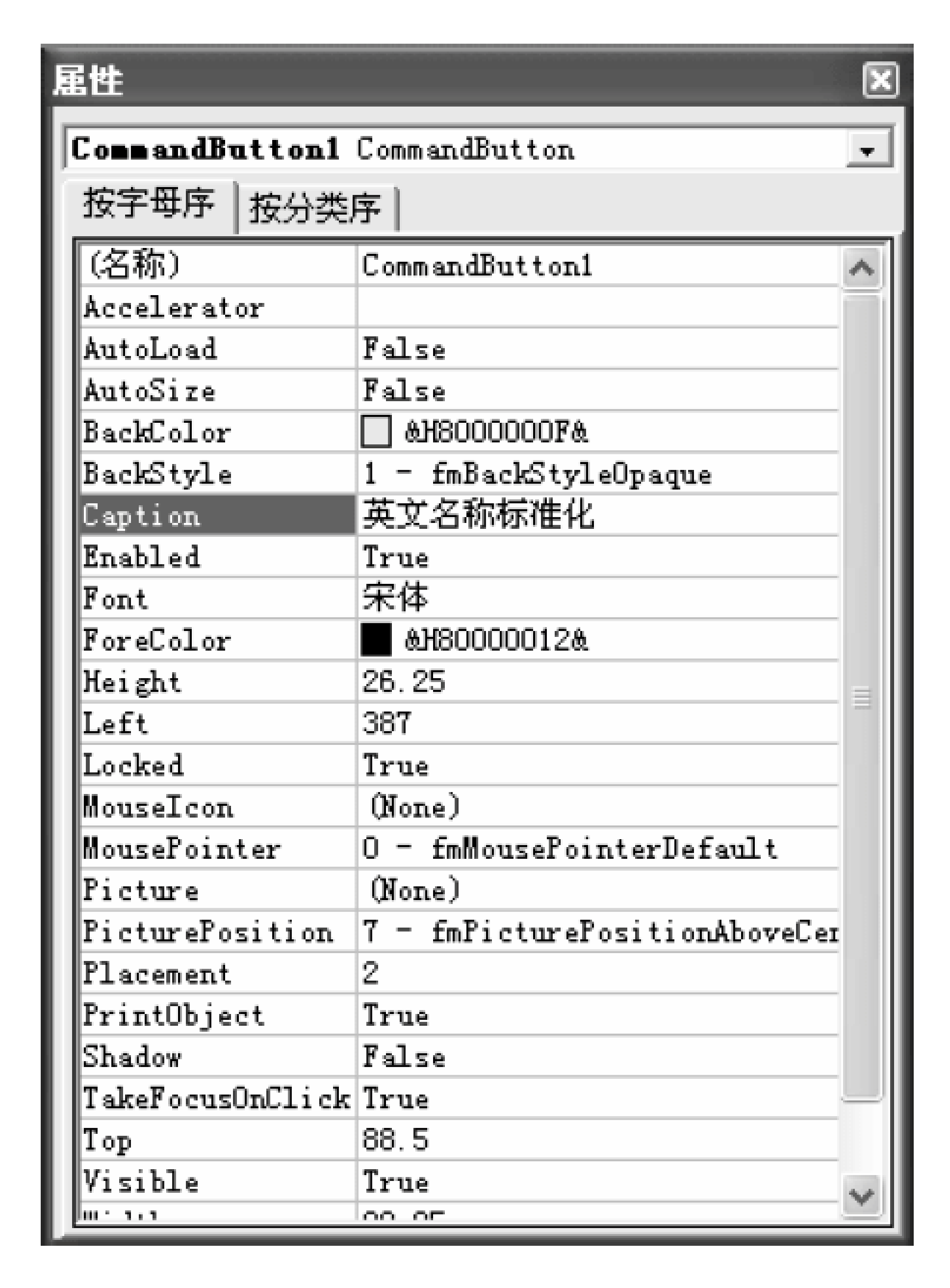


图 11-3　修改命令按钮属性

04 右击【英文名称标准化】按钮，在弹出的快捷菜单中选择【查看代码】菜单命令，即可进入该按钮的代码编辑窗口，从中添加如下程序代码。

```
Private Sub CommandButton1_Click()
Dim t As Variant  ' 定义变量
With Selection    ' 选择区域
  For Each t In Selection
     t.Value = StrConv(t.Value, vbLowerCase)
     ' 将所有的字母全部转换为小写字母
     t.Value = StrConv(t.Value, vbProperCase)
     ' 将字符串中每个单词的开头字母转换为大写字母
  Next
End With
End Sub
```

05 代码输入完毕后，返回工作表。为使工作表的整体布局更加规范，这里首先选择第 2 行，然后在【开始】选项卡中的【单元格】选项区中，选择【插入】➤【插入工作表行】菜单命令，即可插入一行空白单元格，利用鼠标将添加的按钮移到新插入的空白行处，如图 11-4 所示。

员工考核报表

英文名称标准化

编号	姓名	所属部门	业务技能(30)	出勤情况(10)	组织能力(10)	综合成绩
1001	lifang	销售部	25	9	9	91
1002	duanling	财务部	26	10	8	88
1003	chenyan	销售部	24	7	8	80.5
1004	sunshi	销售部	29	8	7	83
1005	zhengjing	财务部	23	9	7	79
1006	yangp	销售部	25	10	8	82
1007	wangyi	销售部	24	10	6	81
1008	liwan	财务部	28	9	6	84
1009	wansi	行政部	20	8	6	71
1010	wanse	后勤部	21	10	6	75
1011	chensan	财务部	26	9	7	83
1012	zanlen	财务部	25	8	7	79
1013	xubin	销售部	28	10	8	89.5
1014	guotou	人事部	27	8	9	87
1015	sinna	销售部	24	9	8	80.5

Sheet1 Sheet2 Sheet3

图 11-4　将命令按钮移至新插入的行中

06 此时该按钮仍处于设计模式，用户可在【开发工具】选项卡的【控件】选项区中单击【设计模式】按钮，即可退出设计模式。

07 选择需要转化的单元格或单元格区域，这里选择“B4：B19”单元格区域，单击【英文名称标准化】按钮即可快速实现标准化转化功能，如图 11-5 所示。

员工考核报表

英文名称标准化

编号	姓名	所属部门	业务技能(30)	出勤情况(10)	组织能力(10)	综合成绩
1001	Lifang	销售部	25	9	9	91
1002	Duanling	财务部	26	10	8	88
1003	Chenyan	销售部	24	7	8	80.5
1004	Sunshi	销售部	29	8	7	83
1005	Zhengjing	财务部	23	9	7	79
1006	Yangp	销售部	25	10	8	82
1007	Wangyi	销售部	24	10	6	81
1008	Liwan	财务部	28	9	6	84
1009	Wansi	行政部	20	8	6	71
1010	Wanse	后勤部	21	10	6	75
1011	Chensan	财务部	26	9	7	83
1012	Zanlen	财务部	25	8	7	79
1013	Xubin	销售部	28	10	8	89.5
1014	Guotou	人事部	27	8	9	87
1015	Sinna	销售部	24	9	8	80.5
1016	Wanyin	人事部	26	10	7	83

图 11-5　实现英文名称标准化

程序中的 StrConV 函数不是 Excel 2010 的内置函数，因此在使用之前用户可以首先检查是否有可用的 StrConV 项。具体的操作步骤如下。

08 在 VBA 代码窗口中，选择【视图】➤【对象浏览器】菜单命令，或按 F2 键，即可打开【对象浏览器】窗口。

09 在【搜索文字】下拉列表框中输入“strconv”后，按下 Enter 键即可进行搜索，搜索结果如图 11-6 所示。

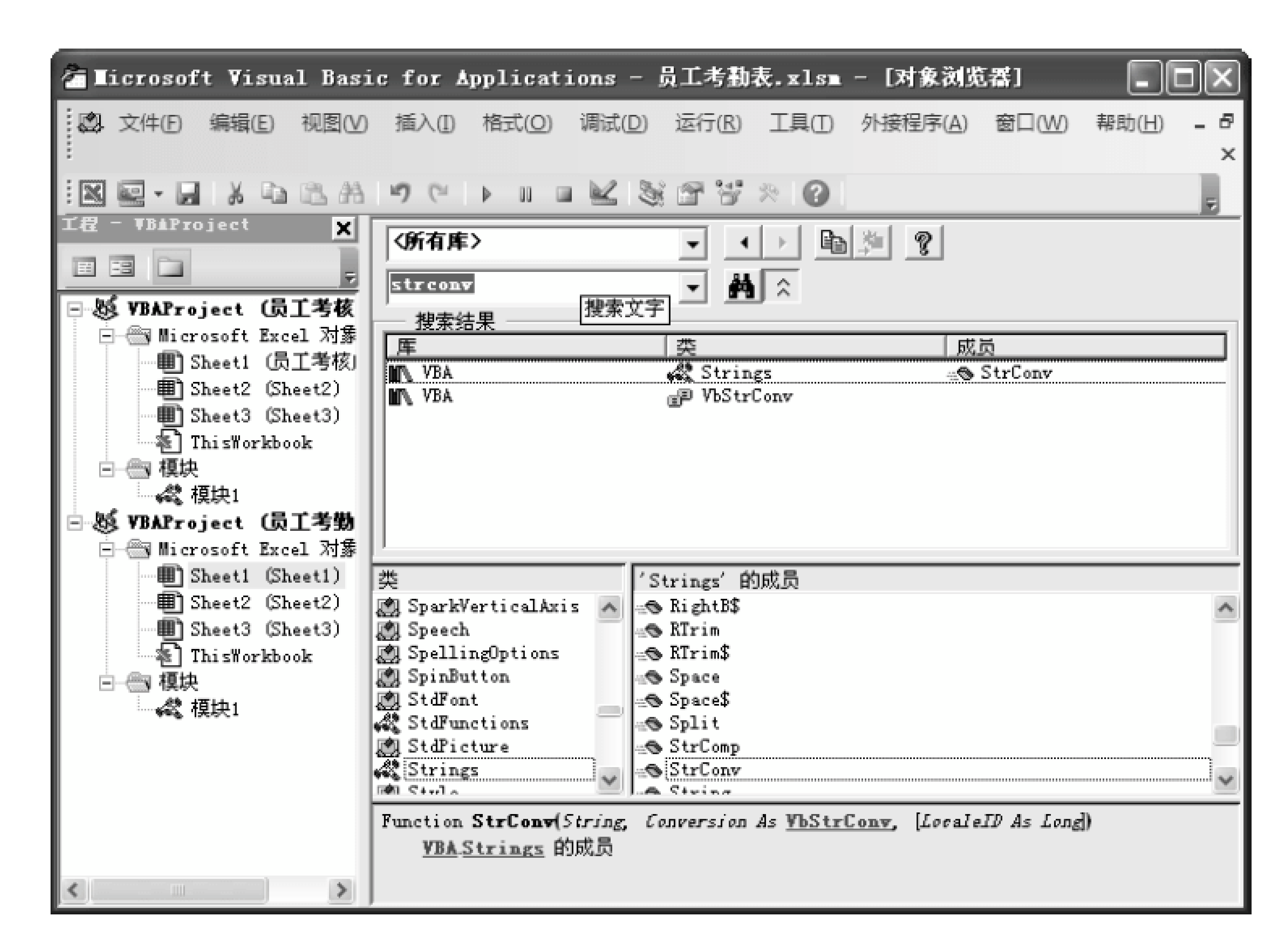


图 11-6　【对象浏览器】窗口

VBA 提供的 StrConV 函数可以将不标准的数据转换为标准格式，其调用格式为：

```
Result=StrConV(strInput,iConversion)
```

该语句的转换结果就是 StrConV 函数的返回值 Result。其中参数 strInput 代表需要进行转换的字符串。参数 iConversion 代表转换后的标准格式。

有关 iConversion 参数的设置值及功能说明如表 11-1 所示。

表 11-1　iConversion 参数的设置值及功能说明

iConversion 参数设置值	功能说明
vbUpperCase	将字符串中的字母全部转换为大写字母
vbLowerCase	将字符串中的字母全部转换为小写字母
vbProperCase	将字符串中每个单词的开头字母转换为大写字母
vbUnicode	使用系统默认代码页将字符串转换为 Unicode
vbFromUnicode	使用系统默认代码页将 Unicode 转换为字符串

11.1.2　转换货币数字为中文形式

在实际工作中，为了数据安全，通常需要将数字款项转换为中文形式后，再进行转账或记录等操作。

【例 11.2】将已有表格中的货币数字转换为中文表示形式，具体操作步骤如下。

这里需要使用 Function 自定义一个循环嵌套函数 standard，然后对其进行调用来实现货币数字到中文形式的转换。

01 按照上节的方法，在工作表中添加控件按钮，并在其【属性】对话框中的【Caption】文本框中输入“货币形式标准化”，修改添加的命令按钮的显示文本，如图 11-7 所示。



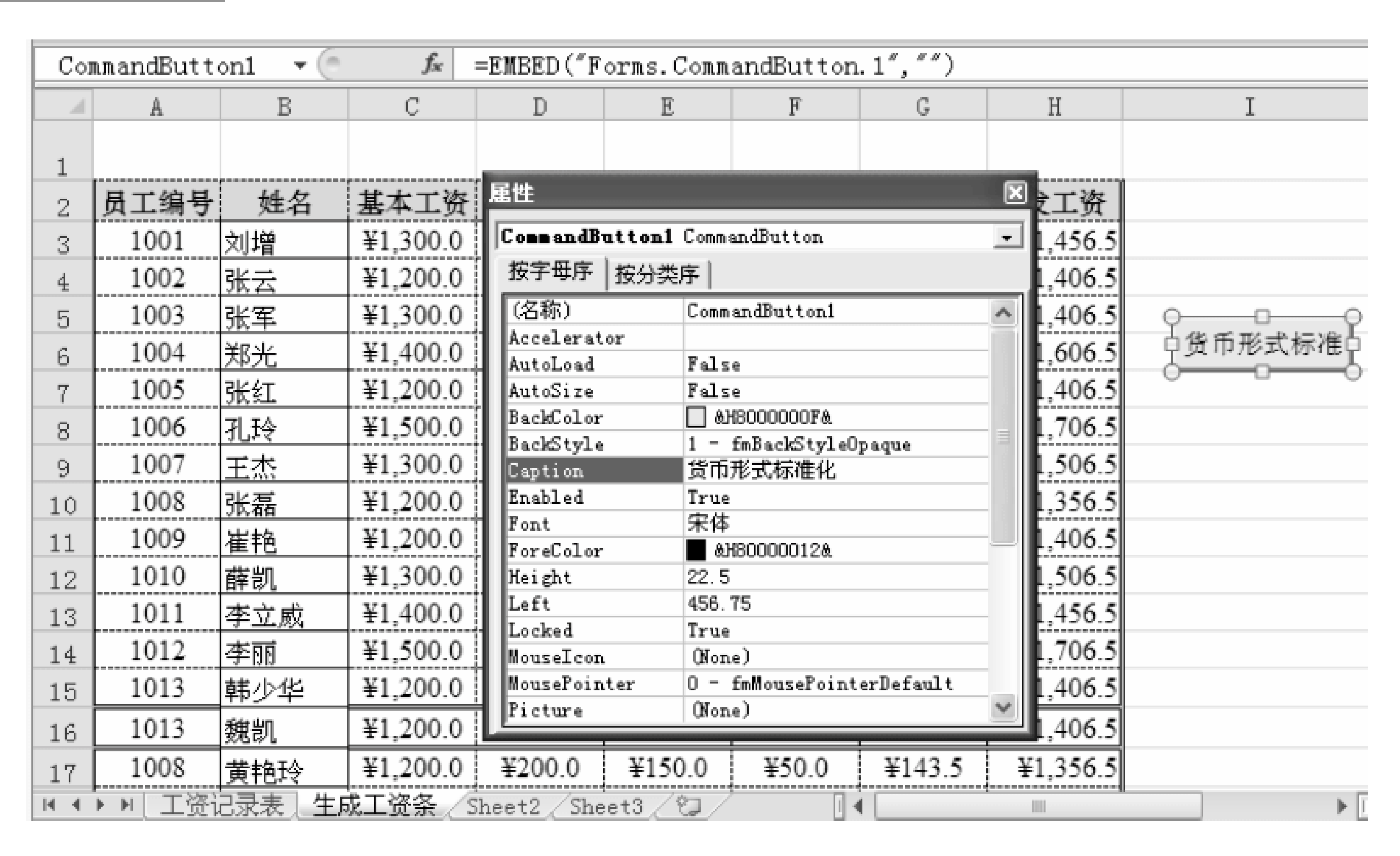


图 11-7　修改按钮属性

02 在设计模式下，双击添加的【货币形式标准化】按钮，进入该按钮的代码编辑窗口，在代码窗口中添加如下的程序代码：

```
    Function standard(ByVal money As Long, ByVal setunit As Integer, time As Integer)
As String
    On Error GoTo error
    Dim danwei As Variant, shuzhi As Variant   '字义用于存储单位和数值的两个数组变量
    Dim dwx As Integer, szx As Integer       '定义记录上述两个数组下标的整型变量
    Dim count As Long   '定义一个用来存储货币数字的长整型变量 count
    Dim str As String   '定义用来存储转换结果的字符串变量
    str = ""
    danwei = Array("", "拾", "佰", "仟", "万", "亿")
    shuzhi = Array("", "壹", "贰", "叁", "肆", "伍", "陆", "柒", "捌", "玖")

    If (Not IsNumeric(money)) And (Len(money) > 10) Then
          standard = "error"
          Exit Function
    End If
    count = Int(Abs(money))   '利用 Abs 函数获取绝对值并将其转换为整型赋给 count
    If count > 0 Then        '对货币数字进行逐位判断
        szx = count Mod 10
        count = Int(count / 10)
        dwx = setunit
        setunit = setunit + 1
       If setunit > 4 Then       '当最低单位量超过万时重新计数
         time = time + 1
         If time = 2 Then      '设置单位亿的判断条件
           dwx = setunit
         End If
        setunit = 1
       End If
       If count > 0 Then
         If (shuzhi(szx) <> "") Then
```

```
            str = standard(count, setunit, time) & shuzhi(szx) & danwei(dwx) & str
          ElseIf (shuzhi(szx) = "") And (dwx = 4) Then
            str = standard(count, setunit, time) & danwei(dwx) & str
          Else
            str = standard(count, setunit, time) & str
         End If
       Else
          str = shuzhi(szx) & danwei(dwx) & str
       End If
    End If
    standard = str  '将转化后的字符串赋值给自定义函数
    Exit Function  '提前退出该函数，以防止在系统没有错误时进入错误处理程序
    error:        '一个错误处理程序，当系统发现错误时就自动跳转执行这部分后面的语句
       MsgBox "系统错误"
    End Function
    Private Sub CommandButton2_Click()
    '对自定义函数 standard 进行调用，“CommandButton2_Click”是控件名
       Dim huobi As Range
       For Each huobi In Range("H3:H22")
          huobi.Offset(0, 1).Value = standard(huobi.Value, 0, 0) & "元"
       Next
    End Sub
```

03 返回工作表，在【开发工具】选项卡的【控件】选项区中单击【设计模式】按钮，退出设计模式。然后选择“H3:H22”单元格区域后，单击【货币形式标准化】按钮，即可将所选单元格区域中的货币数字转换为中文形式，如图 11-8 所示。

I3　壹仟肆佰伍拾陆元

	A	B	C	D	E	F	G	H	I
1	货币形式标准								
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
3	1001	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟肆佰伍拾陆元
4	1002	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
5	1003	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟肆佰陆元
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹仟陆佰陆元
7	1005	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
8	1006	孔玲	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
9	1007	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
10	1008	张磊	¥1,200.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,356.5	壹仟叁佰伍拾陆元
11	1009	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
12	1010	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
13	1011	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰伍拾陆元
14	1012	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
15	1013	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	魏凯	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元

工资记录表　生成工资条　Sheet2　Sheet3

图 11-8　将货币数字转换为中文形式

【提示】本实例中的程序只考虑了位数不超过 10 的数字，如果数字超过 10 位，则会造成数据的溢出。

在本实例的程序设计中涉及了 IsNumeric、Offset 等函数，下面就来具体介绍一下。

（1）IsNumeric 函数的作用是返回 Boolean 值，用来指出表达式的运算结果是否为数字。

其语法格式为：

```
IsNumeric (expression)
```

其中参数 expression 是一个 Variant 变量，它可以包含数值表达式或字符串表达式。当整个 expression 的运算结果为数字时，函数的返回值为 True，否则返回 False。

（2）Len 函数的功能是返回字符串中的字符长度。其语法格式为：

```
Len(text)
```

其中参数 text 表示要查找其长度的文本。

（3）Offset 函数的作用是根据指定的偏移量返回相对于某个单元格区域的另一个单元格区域，其语法格式为：

```
Offset(row,column)
```

其中参数 row 和 column 分别为行偏移量和列偏移量。当它们的取值为正数时则表示向下和向右的偏移量，若为负值则表示向上和向左的偏移量。

（4）Abs 函数的功能是返回数字的绝对值，其语法格式为：

```
Abs(number)
```

其中参数 number 表示需要计算其绝对值的数值。

11.1.3 实现冻结窗格功能

如果原始数据表中含有大量的数据，当用户要查看下面的信息时，需要在工作表中拖动最右侧的滚动条，与此同时每一列的列标也被移动隐藏了，此时设置“冻结窗格”可以将标题冻结，使其不再随着滚动条而移动。

【例 11.3】使用 Excel 2010 VBA 对工作表中的固定区域实现冻结窗格的操作并不复杂，下面就来设置添加控件按钮来实现此功能，具体的操作步骤如下。

01 依照上文介绍的方法，在工作表中添加控件按钮，并在其【属性】对话框中的【名称】文本框中输入“冻结窗格”，这样便于当工作表中 VBA 过程较多时查看程序，可将过程的名称设置为与添加按钮的名称相同；在【Caption】文本框中也输入“冻结窗格”，即可修改添加的命令按钮的显示文本，如图 11-9 所示。



图 11-9　设置按钮属性

02 在设计模式下，双击添加的【冻结窗格】按钮，进入该按钮的代码编辑窗口，在代码窗口中添加如下的程序代码。

```
Private Sub 冻结窗格_Click()
With Selection
ActiveWindow.FreezePanes = True
End With
End Sub
```

【提示】该代码的功能是根据用户选择的区域，将 Window 对象的 FreezePanes 属性值设置为 True，即设置冻结窗格。这样当用户选择某单元格后，单击【冻结窗格】按钮即可实现冻结窗格功能。

03 返回工作表，在【开发工具】选项卡的【控件】选项区中单击【设计模式】按钮，即可退出设计模式。然后选择 G5 单元格后，单击【冻结窗格】按钮，即可看到冻结窗格的效果，此时当向下移动垂直滚动条时会发现前 4 行是固定不动的；当向右拖动水平滚动条时，会发现 A、B、C、D、E、F 列是固定不动的，如图 11-10 所示。

	A	B	C	D	E	F	G	H	I
1	货币形式标准		冻结窗格						
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
3	1001	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟肆佰伍拾陆元
4	1002	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
14	1012	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
15	1013	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	魏凯	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	黄艳玲	¥1,200.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,356.5	壹仟叁佰伍拾陆元
18	1009	陈晓娟	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
19	1010	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
20	1011	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰伍拾陆元
21	1012	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
22	1013	张克歌	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
23									
24									

图 11-10　实现冻结窗格

04 如果想取消冻结窗格的效果，可在工作表中切换到【视图】选项卡，在【窗口】选项区中选择【冻结窗格】➤【取消冻结窗格】菜单命令即可。

【提示】在使用冻结窗格功能时，一般冻结的单元格是所选中单元格之前的列和之上的行。例如，选中单元格 A6，则冻结前 5 行；选中单元格 B1，则冻结前 1 列；如果需要同时冻结行和列，则可选中行和列交叉处的单元格，例如选中 B7，则冻结的是前 6 行和第 1 列的单元格区域。



11.2 对产品进行筛选与排序

在日常办公中，经常需要对一些表单进行数据的分析与统计，此时可以使用 Excel 2010 提供的筛选和排序功能。本节将详细介绍利用 VBA 代码实现数据分析与统计等功能的操作方法。

11.2.1 实现产品自动筛选

用户可以使用 Excel 2010 VBA 程序实现自动筛选功能，从而在工作表中查看符合特定条件的所有记录。

【例 11.4】利用 VBA 筛选出“实发工资”在 1500 元以上的员工，具体操作步骤如下。

01 参照前面介绍的方法，在工作表中添加控件按钮，并在其【属性】对话框的【按字母序】选项卡中，修改【名称】和【Caption】的属性值，如图 11-11 所示。

图 11-11　修改【筛选】按钮属性值

02 在设计模式下，双击添加的【筛选】按钮，进入该按钮的代码编辑窗口，在代码窗口中添加如下的程序代码：

```
Private Sub 筛选_Click()
Selection.AutoFilter    '设置自动筛选
Selection.AutoFilter Field:=8, Criteria1:=">1500"     '设置筛选条件
End Sub
```

代码分析：利用 AutoFilter 属性可实现自动筛选功能；Field:=8 表示工作表中的第 8 列；

Criteria1:=">1500"表示筛选大于 1500 的记录。

03 返回工作表，在【开发工具】选项卡的【控件】选项区中单击【设计模式】按钮，即可退出设计模式。然后单击【筛选】按钮即可筛选出工资大于 1500 的员工，如图 11-12 所示。

	A	B	C	D	E	F	G	H	I
1	货币形式标准		冻结窗格		筛选				
2	员工编	姓名	基本工	住房补	满勤奖	事假扣	代扣保	实发工资	
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹仟陆佰陆元
8	1006	孔玲	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
9	1007	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
12	1010	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
14	1012	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
19	1010	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
21	1012	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟柒佰陆元
23									
24									
25									

图 11-12　自动筛选工资大于 1500 的员工

04 运行完筛选程序后，如果想取消筛选功能，在工作表的【数据】选项卡的【排序和筛选】选项区中单击【筛选】按钮，即可取消筛选操作。

11.2.2 实现产品自动排序

在分析数据时常常会应用到自动排序功能。合理的排序可以方便查看数据内容。

【例 11.4】对“员工工资表”实施自动排序，依据“实发工资”进行由高到低排序，具体操作步骤如下。

01 参照前面介绍的方法，在工作表中添加控件按钮，并在其【属性】对话框的【按字母序】选项卡中，修改【名称】和【Caption】的属性值均为“排序”，如图 11-13 所示。



图 11-13　修改【排序】按钮属性

02 在设计模式下，双击添加的【排序】按钮，进入该按钮的代码编辑窗口，在代码窗口中添加如下的程序代码：

```
    Private Sub 排序_Click()
    Range("B3:H22").Sort        Key1:=Range("H3"),        Order1:=xlDescending, 
Header:=xlGuess
    End Sub
```

03 返回工作表，在【开发工具】选项卡的【控件】选项区中单击【设计模式】按钮，即可退出设计模式。然后单击【排序】按钮，即可实现“实发工资”为依据的降序排列功能，如图 11-14 所示。

	A	B	C	D	E	F	G	H	I
1		货币形式标准	冻结窗格		筛选	排序			
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
3	1001	孔玲	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰伍拾
4	1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
5	1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹仟陆佰陆元
7	1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟肆佰陆元
8	1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟柒佰陆元
9	1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
10	1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟叁佰伍拾
11	1009	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰陆元
12	1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾
14	1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
15	1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾

工资记录表 / 生成工资条 / Sheet2 / Sheet3

图 11-14　实现降序排列功能

在该程序代码中首先选择排序的区域“B3:H22”，然后对该区域执行 Sort 方法，即实现自动排序。这里按“实发工资”降序排列，如果需要按照其他字段排序，则只需要修改代码中的“H3”即可。



11.3 对产品信息进行管理

在 Excel 2010 中，管理产品信息的方法有多种。但使用不同的方法其难易程度和处理数据的结果是不同的。下面将分别使用记录单、VBA 程序代码和 VBA 控件 3 种方法实现对公司产品信息的管理，并且利用 VBA 代码实现不同地区的业务比例的计算。

11.3.1 实现记录单管理

在 Excel 2010 中，使用记录单管理数据信息是比较好的方法。记录单中不仅可以显示一条完整的记录信息，同时还可以利用记录单添加、删除、修改和查询任意一条记录信息，其具体的操作步骤如下。

01 在工作表中切换到【数据】选项卡，在【记录单】选项组中单击【记录单】按钮，即可打开【生成工资条】记录单，从中可以看到记录单中显示了当前的记录位置、工作表中各字段名称以及各字段的对应值等信息，如图 11-15 所示。



图 11-15　Sheet1 记录单

02 单击【新建】按钮，该记录单处于“新建记录”状态，分别在与各字段名称对应的文本框中输入添加的信息，如图 11-16 所示。



图 11-16　输入新建记录单的信息

03 输入完毕后，按下 Enter 键，即可将新建的记录信息添加到工作表中，如图 11-17 所示。

员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
1012	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,806.5	壹仟肆佰伍拾
1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹仟陆佰陆元
1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟肆佰陆元
1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟柒佰陆元
1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟叁佰伍拾
1009	李立成	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰陆元
1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾
1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾

图 11-17　显示新建的记录信息

04 如果想要删除某条记录，则可以在打开的记录单对话框中通过单击【上一条】或【下一条】按钮，找到需要删除的记录信息，然后单击【删除】按钮，此时系统则弹出一个确认删除的提示信息，单击【确定】按钮即可将所选记录删除，如图 11-18 所示。



图 11-18　删除提示信息

05 若要查看某条指定的记录，可以单击【条件】按钮，在弹出的空白文本框中输入查询的条件，例如在【员工编号】文本框中输入"1005"，如图 11-19 所示。

06 输入完毕后，按下 Enter 键即可得到查询结果。当有多个查询结果时，可以通过单击【上一条】或【下一条】按钮进行逐条查看，如图 11-20 所示。



图 11-19　输入查询条件



图 11-20　显示查询结果

11.3.2 VBA 程序代码管理

使用 VBA 程序代码同样可以实现与上述记录单操作相同的功能，如添加、删除、修改和查询等。但是如果使用 VBA 实现以上功能的话会很麻烦，并不能体现 VBA 的高效办公特性。

但是使用 VBA 查询客户资料与记录单相比，还是有一定优势的，优点是能够直观地在工作表中显示出查询结果，从而可以在工作表中直接对查询结果进行修改或删除等操作。操作水平比较高的读者可以考虑使用该方法。

【例 11.5】使用 VBA 命令实现对现有工作表的信息管理，具体操作步骤如下。

为了实现 VBA 程序代码的快速启动，这里首先设置 VBA 程序代码的启动方式，即插入一张图片，只要单击设置的启动图片，即可执行相应的程序代码。

01 在 Excel 2010 工作表的功能区选择【插入】选项卡，在【插图】选项组中单击【剪贴画】按钮，打开【剪贴画】对话框，如图 11-21 所示。



图 11-21　【剪贴画】对话框

02 从中单击要插入的剪贴画，即可在工作表中插入所选中的图片。利用鼠标调整图片的大小，并将其置于工作表合适的位置，如图 11-22 所示。

	A	B	C	D	E	F	G	H	I
1	货币形式标准		冻结窗格		筛选	排序			
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保	[illegible]	[illegible]
3	1012	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥143.	[illegible]	肆佰伍拾
4	1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.	[illegible]	肆佰陆元
5	1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.	[illegible]	佰陆元
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥14	[illegible]	佰陆元
7	1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	[illegible]	肆佰陆元
8	1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	[illegible]	壹仟柒佰陆元
9	1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
10	1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟叁佰伍拾
11	1009	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰陆元
12	1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾
14	1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
15	1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾

工资记录表 / 生成工资条 / Sheet2 / Sheet3

图 11-22　插入剪贴画并调整其大小及位置

03 右击插入的图片，在弹出的快捷菜单中选择【指定宏】菜单命令，打开【指定宏】对话框，如图 11-23 所示。



图 11-23　【指定宏】对话框

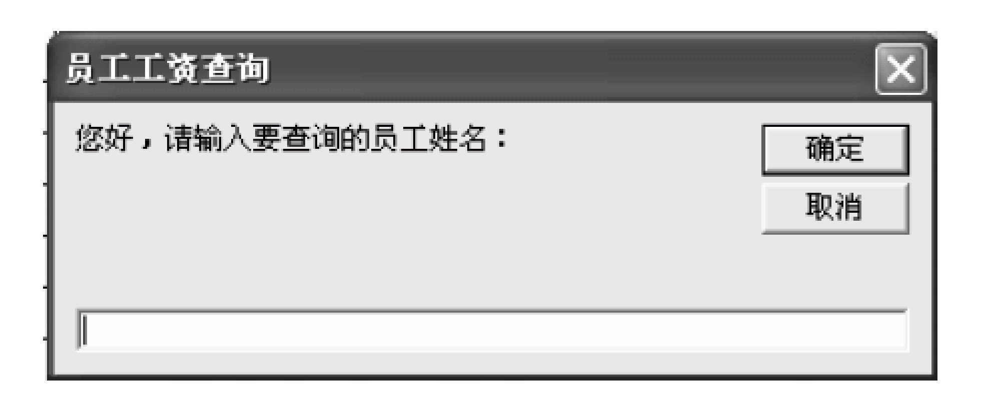
04 在文本框中输入“员工工资查询”后，单击【新建】按钮，即可在 VBA 代码窗口中插入相应的模块。在该模块中输入如下 VBA 代码程序：

```
Sub 产品信息查询 ()
Dim i As Integer
Dim Sellarea As String   ' 定义用于保存输入信息的字符串变量
Dim Rowcount As Integer   ' 定义用于记录行数的整型变量
Dim CunZai As Boolean   ' 定义一个用于判断查询的地区是否存在的布尔变量
    Rowcount = ActiveSheet.[B3].CurrentRegion.Rows.count
    ' 把当前工作表的行数赋值 iCount
    Sellarea = InputBox("您好，请输入要查询的员工姓名：", "员工工资查询")
    ' 让用户输入信息
    For i = 3 To Rowcount          ' 3 表示从第三行开始查询
        If Cells(i, 2).Value = Sellarea Then      ' 2 表示查询第二列
            CunZai = True
          MsgBox "查询的员工信息已找到！", vbOKOnly & vbInformation, "查询结果"
```

```
            Rows(i).Select   ' 选择员工信息所在的行
            Selection.Interior.ColorIndex = 43   '将查询结果所在行填充颜色设置为绿色
            ' Sheets("Sheet1").Select
            Exit For              '退出 For 循环
        End If
        Selection.Interior.ColorIndex = 0'当再次运行该程序时取消当前选中区域的填充效
果
     Next i
   If CunZai = False Then
      MsgBox "没有此员工工资信息，请重新核实！",vbOKOnly & vbExclamation,"没有找到！
"
   End If
   End Sub
```

05 输入完毕后返回工作表，单击其他任意一个单元格，即可使添加的剪贴画退出编辑状态。此时将鼠标指针移到该剪贴画上，待鼠标指针变成时，单击鼠标左键，即可运行指定的 VBA 程序代码，并弹出【员工工资查询】对话框，如图 11-24 所示。

06 在文本框中输入要查询的员工姓名（如“张云”）后，单击【确定】按钮，系统即可弹出【查询结果】提示信息，如图 11-25 所示。



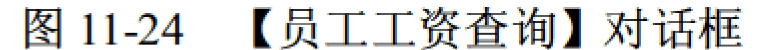


图 11-24　【员工工资查询】对话框

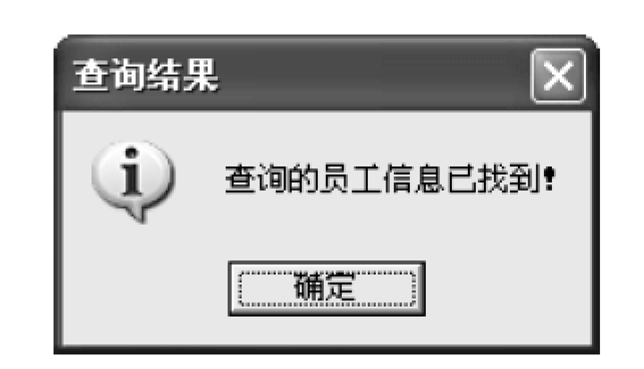


图 11-25　【查询结果】对话框

07 单击【确定】按钮，即可在工作表中以绿色显示出所查询的地区，如图 11-26 所示。

	A	B	C	D	E	F	G	H	I
1	货币形式标准		冻结窗格		筛选	排序			
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保[illegible]	[illegible]	
3	1012	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥143.[illegible]	[illegible]	[illegible]肆佰伍拾陆元
4	1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143[illegible]	[illegible]	[illegible]肆佰陆元
5	1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143[illegible]	[illegible]	[illegible]佰陆元
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥14[illegible]	[illegible]	[illegible]佰陆元
7	1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	[illegible]	[illegible]肆佰陆元
8	1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	[illegible]	壹仟柒佰陆元
9	1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
10	1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟叁佰伍拾陆元
11	1009	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰陆元
12	1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾陆元
14	1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
15	1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾陆元
18	1009	魏凯	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
19	1010	陈晓娟	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟伍佰陆元
20	1011	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

工资记录表　生成工资条　Sheet2　Sheet3

图 11-26　以绿色显示所查询的结果

08 当工作表中没有所查询的地区记录时，系统会弹出【没有找到】对话框，提示用户不存在所查询的记录，请重新核实，如图 11-27 所示。



图 11-27　没有找到提示信息

11.3.3　VBA 控件查询信息

除了使用 VBA 程序命令实现信息查询外，还可以使用 VBA 控件来查询相关信息，具体内容介绍如下。

添加控件实现通过员工姓名查询员工实发工资，具体的操作步骤如下。

01 为了便于对单元格区域的引用，首先对 ˝B2：C22˝ 单元格区域进行定义。右击选中的 ˝B2：C22˝ 单元格区域，在弹出的快捷菜单中选择【定义名称】菜单命令，打开【新建名称】对话框，如图 11-28 所示。



图 11-28　【新建名称】对话框

02 在【名称】文本框中输入 ˝查询基本工资˝，单击【确定】按钮，完成对选中单元格区域名称的定义。

03 选择功能区【开发工具】选项卡，在【控件】选项组中单击【插入】按钮，弹出其相应的下拉列表，如图 11-29 所示。

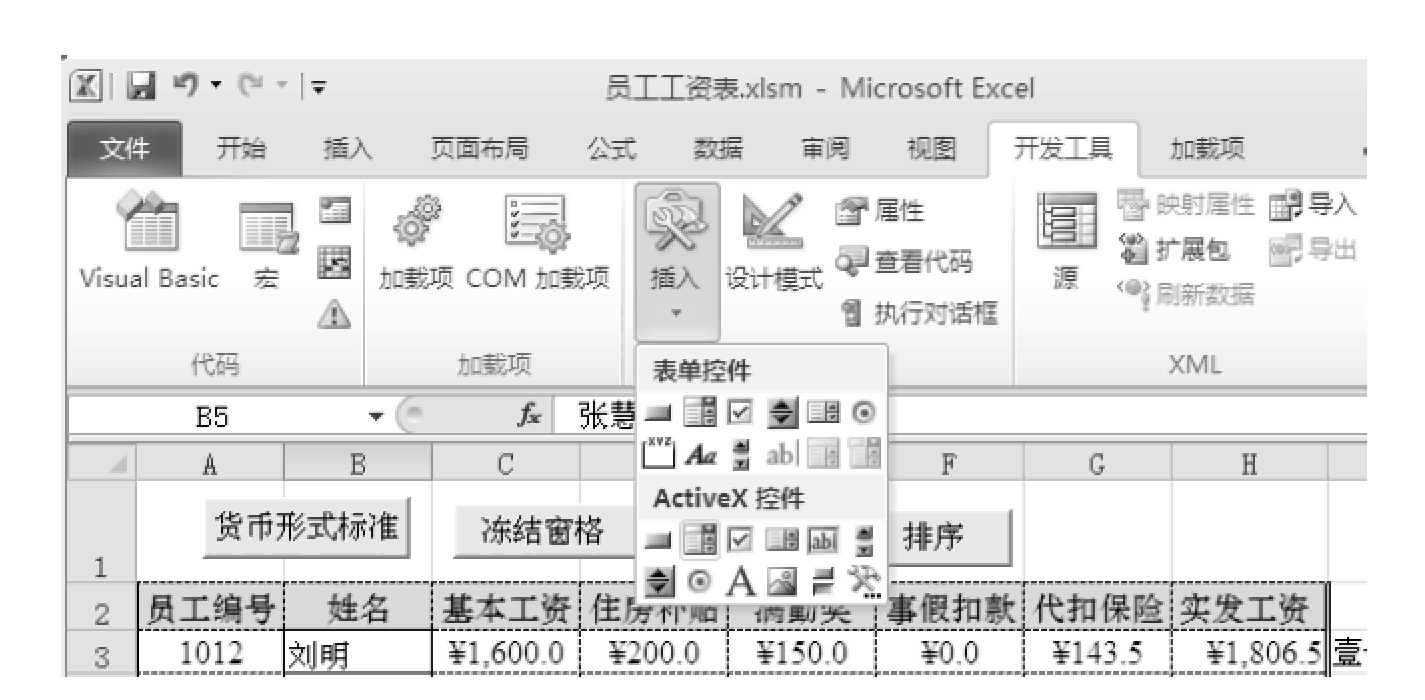


图 11-29　控件列表

04 从中单击【组合框】按钮，在工作表中的合适位置拖放鼠标，添加一个组合框，如图 11-30 所示。

ComboBox1　=EMBED("Forms.ComboBox.1","")

货币形式标准　冻结窗格　筛选　排序

	A	B	C	D	E	F	G	H	I
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
3	1012	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,806.5	壹
4	1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟
5	1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹仟陆
7	1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟肆
8	1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟柒佰陆元
9	1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹仟伍佰陆元
10	1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	壹仟叁佰伍拾陆
11	1009	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟肆佰陆元
12	1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾陆
14	1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
15	1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾陆

工资记录表　生成工资条　Sheet2　Sheet3

图 11-30　插入的组合框

05 右击插入的组合框，在弹出的快捷菜单中选择【属性】菜单命令，打开【属性】窗口，切换到【按字母序】选项卡中，在“Style”选项右侧的下拉列表中选择 2-fmStyleDropDownList 选项，设置用户必须从列表中选择值，如图 11-31 所示。



图 11-31　设置【Style】属性值



06 选择 LinkedCell 选项，在其右侧的文本框中输入“I1”，即选择一个空的单元格 I1 作为显示查询数据结果的地方；在 ListFillRange 选项的右侧文本框中输入定义的单元格区域名称“查询基本工资”，如图 11-32 所示。

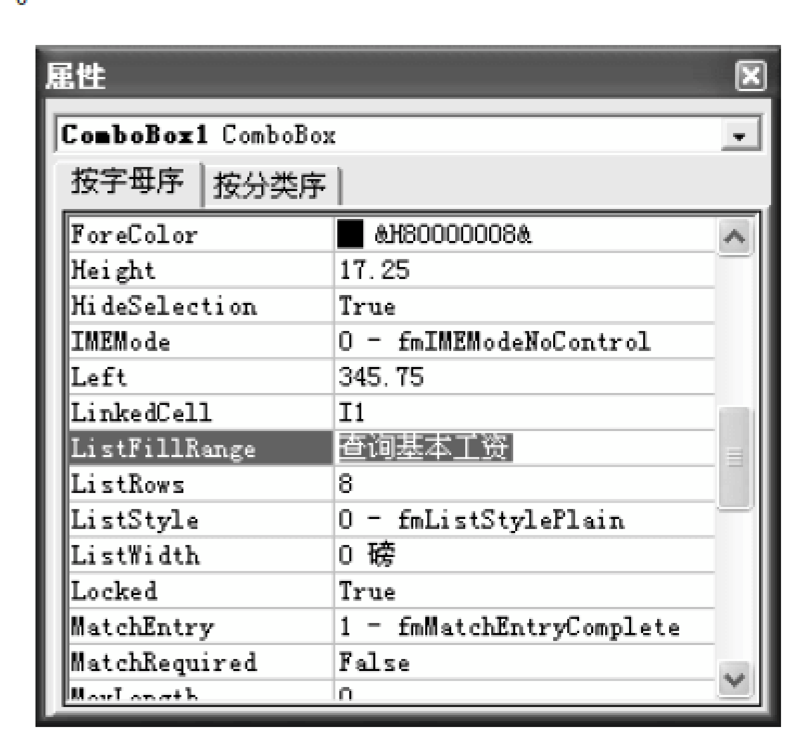


图 11-32　设置查询数据结果的显示位置

07 在 ColumnCount 选项右侧的文本框中设置其值为 2，表示当前引用区域中共有 2 列数据；在 BoundColumn 选项右侧的文本框中设置其值也为 2，表示需要查找的是第 2 列单元格的内容，如图 11-33 所示。

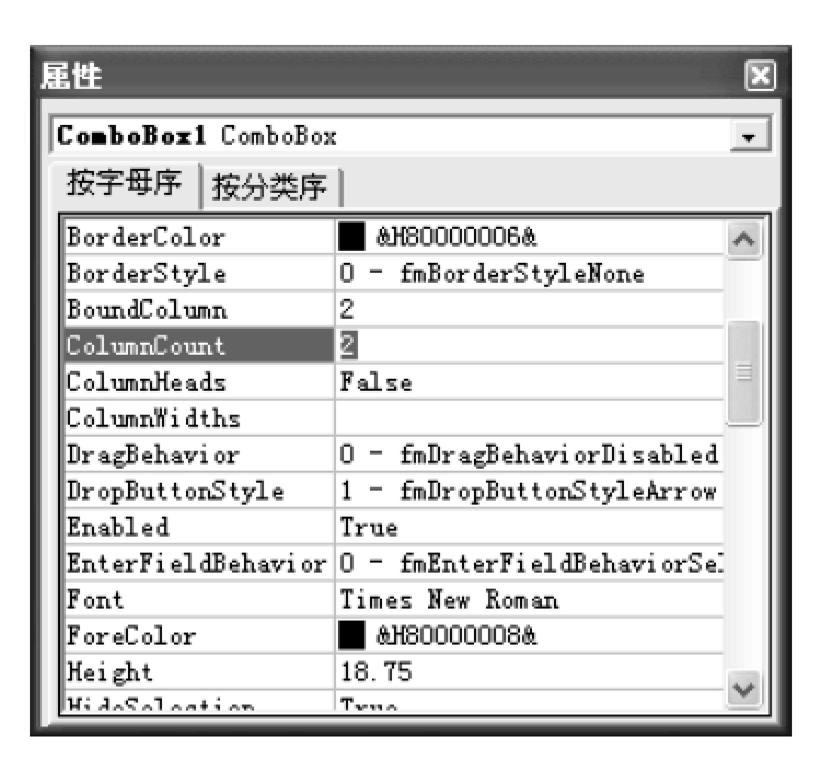


图 11-33　设置 ColumnCount 和 BoundColumn 的属性值

08 单击 Font 选项右侧的【选择】按钮 **...** ，打开【字体】对话框，从中选择字体为 Times New Roman，字形为“常规”，大小为 14，如图 11-34 所示。



图 11-34　【字体】对话框

09 单击【确定】按钮返回工作表，选择功能区【开发工具】选项卡，单击【控件】选项组中【设计模式】按钮，退出设计模式。然后在 I1 单元格中输入定义的单元格区域名称 "查询基本工资"。展开组合框的下拉列表，可显示出员工姓名列表，如图 11-35 所示。

	A	B	C	D	E	F	G	H	I
1		货币形式标准	冻结窗格	筛选	排序				查询基本工资
2	员工编号	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扌	资	
3	1012	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥1	6.5	壹仟肆佰伍拾陆
4	1002	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥1	6.5	壹仟肆佰陆元
5	1003	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥1	6.5	壹仟肆佰陆元
6	1004	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥1	6.5	壹
7	1005	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹
8	1006	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹
9	1007	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹
10	1008	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	
11	1009	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟
12	1010	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	1011	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾陆
14	1012	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元
15	1013	张红	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
16	1013	崔艳	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰陆元
17	1008	韩少华	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟叁佰伍拾陆

姓名
刘明
李丽
张慧杰
郑光
王杰
薛凯
张涛涛

工资记录表　生成工资条　Sheet2　Sheet3

图 11-35　显示员工姓名列表

10 选中任意一名员工，在单元格 I1 中会显示其基本工资，如图 11-36 所示。



	B	C	D	E	F	G	H	I
1	形式标准	冻结窗格	筛选	排序		李丽		1500
2	姓名	基本工资	住房补贴	满勤奖	事假扣款	代扣保险	实发工资	
3	刘明	¥1,600.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,806.5	壹仟肆佰伍拾陆元
4	李丽	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
5	张慧杰	¥1,500.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,706.5	壹仟肆佰陆元
6	郑光	¥1,400.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,606.5	壹
7	王杰	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹
8	薛凯	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	壹
9	张涛涛	¥1,300.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,506.5	
10	刘增	¥1,300.0	¥200.0	¥150.0	¥50.0	¥143.5	¥1,456.5	
11	李立威	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟
12	陈磊	¥1,400.0	¥200.0	¥150.0	¥150.0	¥143.5	¥1,456.5	壹仟伍佰陆元
13	张云	¥1,200.0	¥200.0	¥150.0	¥0.0	¥143.5	¥1,406.5	壹仟肆佰伍拾陆元
14	张军	¥1,300.0	¥200.0	¥150.0	¥100.0	¥143.5	¥1,406.5	壹仟柒佰陆元

图 11-36　显示所选员工的基本工资



11.4 在用户窗体中添加控件

用户窗体是控件的载体，也是建立一个对话框的前提。而且在设计用户窗体时也会经常用到控件，例如标签、文本框等。

11.4.1 用户窗体的基本操作

用户窗体的基本操作如下。

1. 插入、显示及移除窗体

在本节中主要介绍如何插入、显示及移除窗体，具体内容如下。

1）插入窗体

在 VBA 代码窗口中选择【插入】➢【用户窗体】菜单命令，可插入一个空白窗体，插入后的窗体名称（默认为 UserForm1）会出现在工程窗口中，如图 11-37 所示。

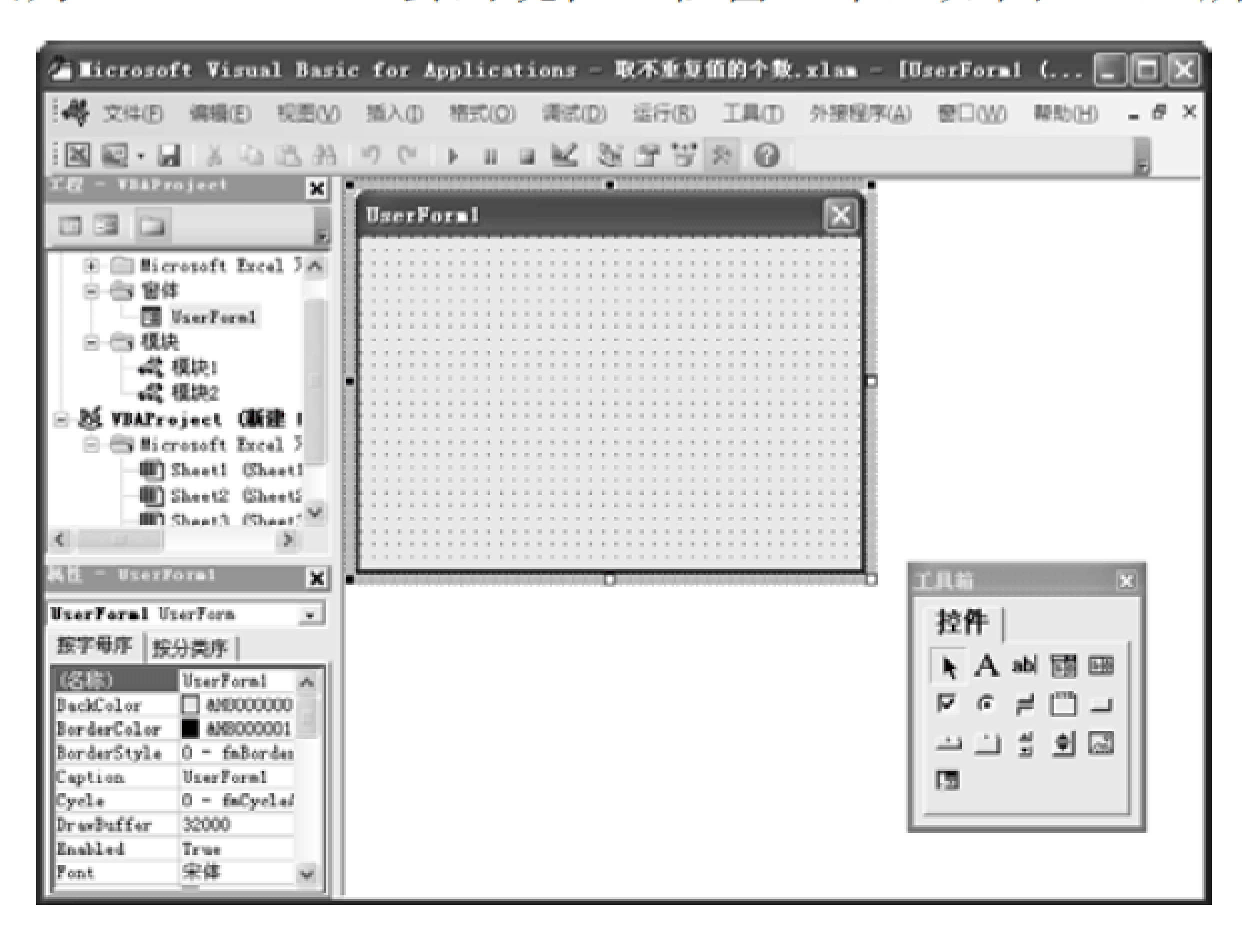


图 11-37　插入的空白窗体

在【工程资源管理器】中右击，在弹出的快捷菜单中选择【插入】➢【用户窗体】菜单命令，同样可以插入一个空白窗体。

2）显示窗体

在 VBA 代码窗口中插入的窗体是在设计模式下，如果需要在窗体中完成一些功能，还必须将窗体脱离设置模式并显示出来，如图 11-38 所示。

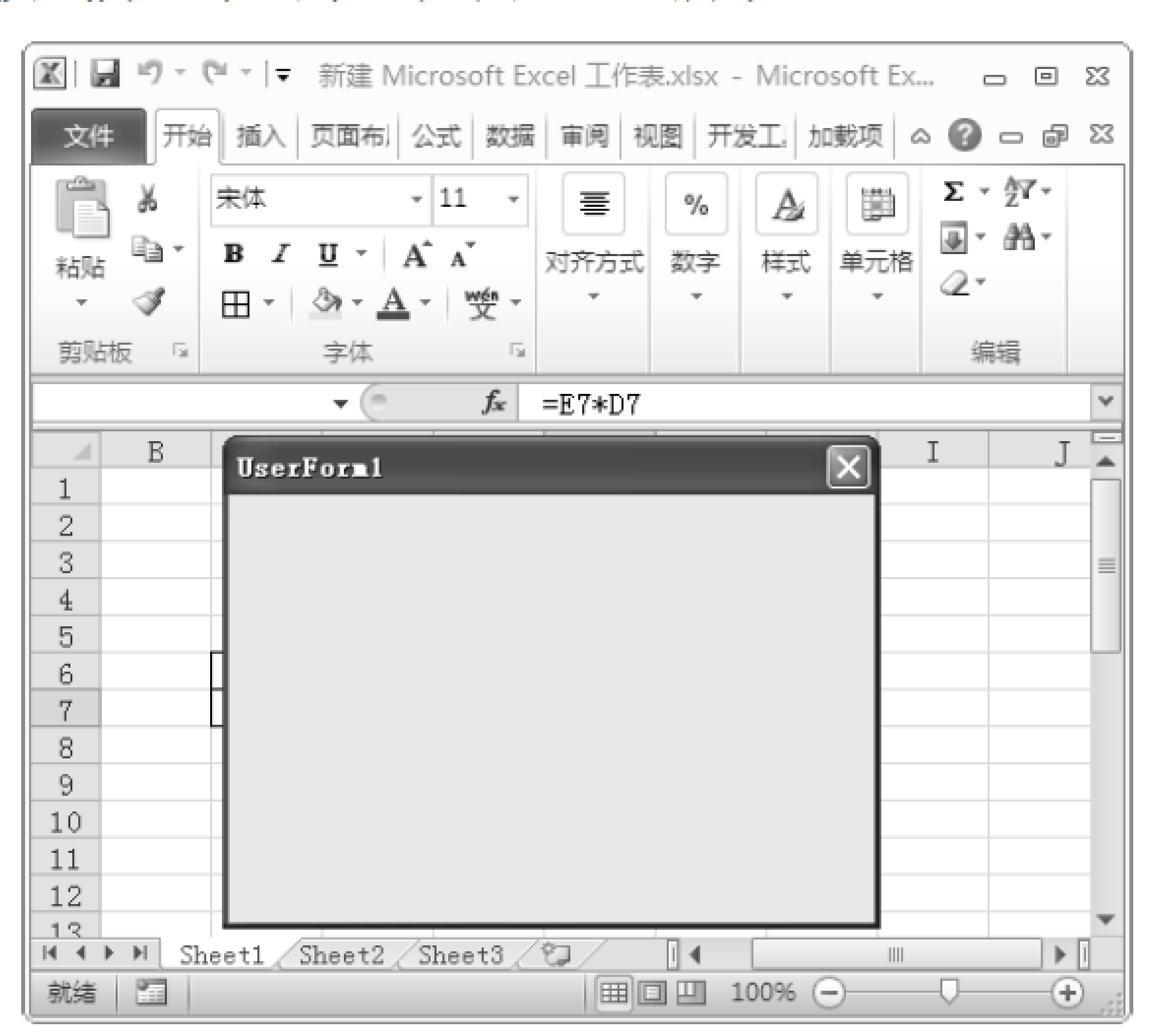


图 11-38　运行后的用户窗体

要运行并显示窗体有如下两种方法：

（1）选择窗体并在常用工具栏中单击【运行子过程/用户窗体】按钮▶。
（2）利用程序代码显示窗体。

添加一个模块并输入下面的代码，运行后可以显示 UserForm1 窗体。

```
Private Sub UserForm_Click()
  UserForm1.Show          ' 窗体的 Show 方法可以显示并运行窗体
End Sub
```

【提示】UserForm1.Show 可添加到程序的任何位置，当程序运行到该句时，便会显示窗体；如果在程序中使用 UserForm1.Show 方法但该窗体并不存在，将出现如图 11-39 所示的错误提示。



图 11-39　错误提示

3）关闭窗体

关闭窗体有两种，一种是手动关闭，即单击窗体上的关闭按钮关闭窗体；另一种是使用代码关闭窗体。

例如，使用代码实现当单击窗体时关闭该窗体，具体的操作步骤如下。

01 在 VBA 代码窗口中选择【插入】➢【用户窗体】菜单命令，插入一个空白窗体，然后双击该窗体，即可打开窗体代码窗口，如图 11-40 所示。

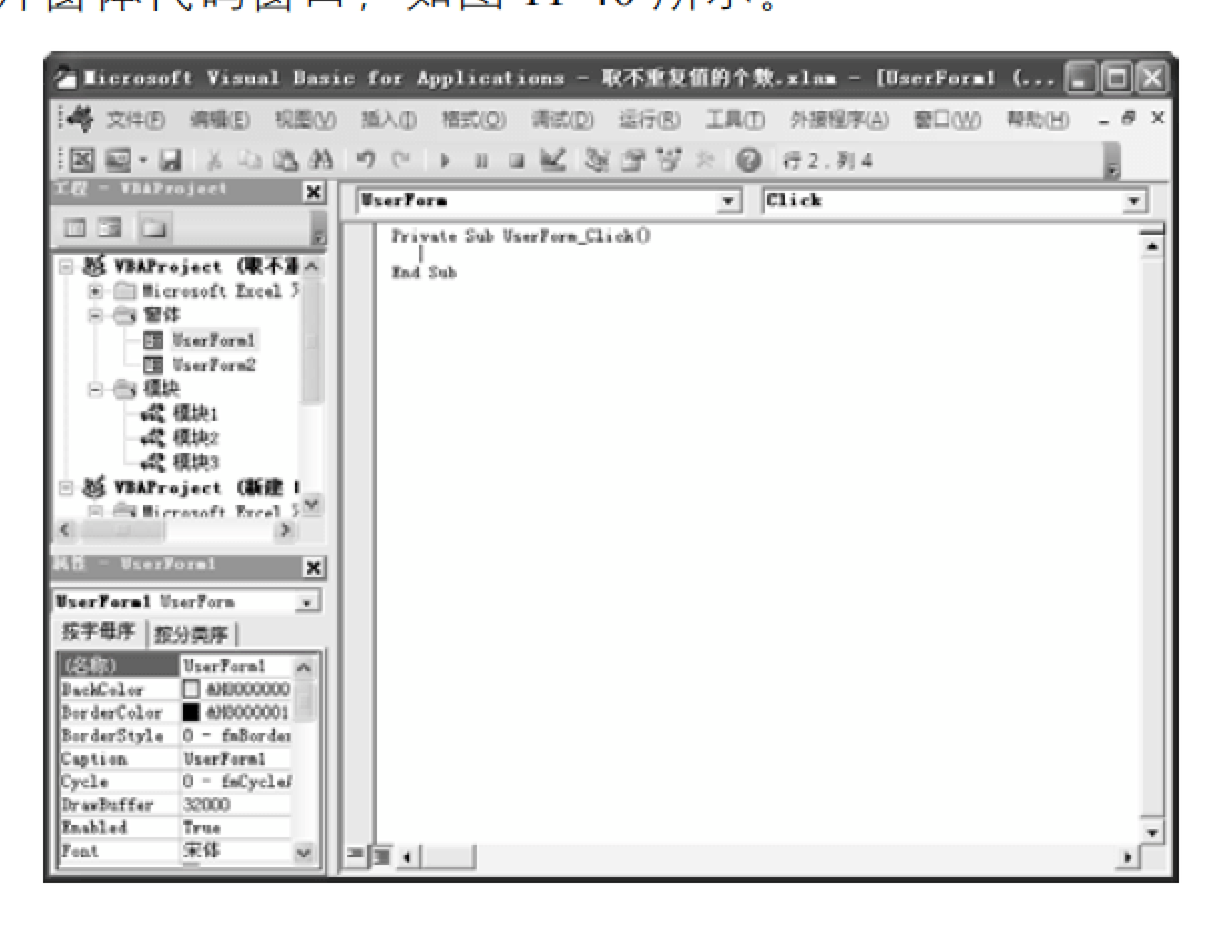


图 11-40　打开窗体代码窗口



02 在代码窗口的代码框架中输入 Unload UserForm1 语句，具体操作命令如下。

```
Private Sub UserForm Click()
  Unload UserForm1
End Sub
```

03 当窗体在运行并处于显示状态时，单击窗体便会关闭窗体。

4）移除窗体

移除窗体的具体操作方法如下。

在【工程资源管理器】窗口中右击要移除的窗体，在弹出的快捷菜单中选择【移除窗体】菜单命令，系统即可弹出如图 11-41 所示的提示信息，如果不需要导出，直接单击【否】按钮，即可将其移除。



图 11-41　移除提示信息

2. 设置窗体特征



图 11-42　修改窗体的名称

1）窗体的名称

新插入的窗体需要用户给窗体命名和修改，具体操作方法如下。

新插入的窗体如果是本工作簿第一次插入的窗体，窗体的默认名称为 UserForm1，第二次插入的窗体则是 UserForm2，再次插入的窗体默认名称以此类推。如果想修改窗体名称，可以在窗体的【属性】窗口中修改属性。在【工程】窗口中的窗体显示名称也会改为在【属性】窗口中修改后的名称，如图 11-42 所示。

窗体的名称可以在程序中直接使用。

```
Sub 显示窗体()
  第一个窗体.Show
End Sub
```

在代码中使用更为确切的窗体名称会有利于代码的维护。如当看到程序中含有 UserForm1 或 UserForm2 时，可能不容易明白正在调用哪个具体的窗体。而“第一个窗体.Show”则可以很清楚地知道程序运行结果是显示第一个窗体。

2）窗体的标题栏文字

窗体的标题栏文字是窗体顶部蓝色区域所显示的文字，通过下面的两种方法可以修改窗体的标题栏。

（1）手工修改窗体属性中的 Caption 属性值，即可把窗体名称修改，如图 11-43 所示。

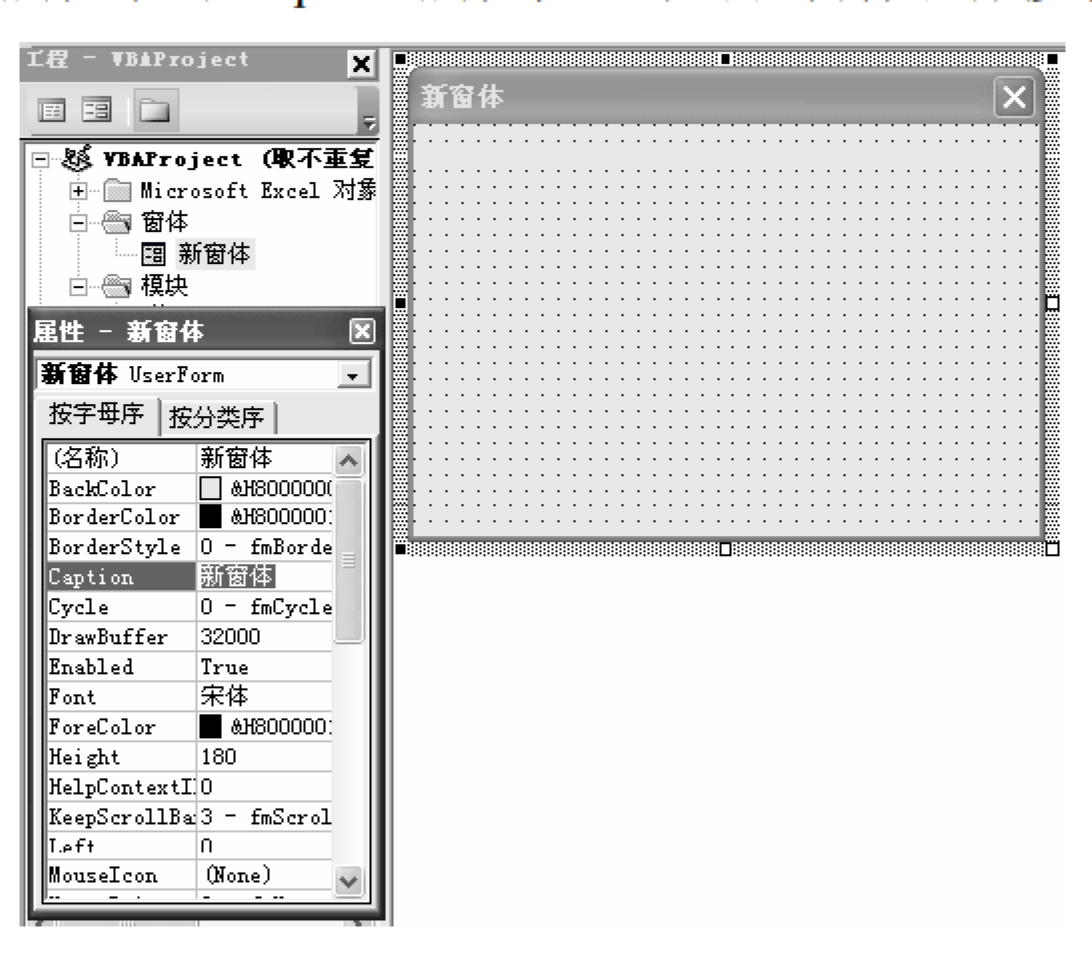


图 11-43　手工设置窗体标题栏文字

（2）在程序中修改窗体名称。

例如，在如图 11-54 所示的工作表中，单击窗体后使窗体的标题栏内容修改为单元格 C3 的内容，则可以在用户窗体代码窗口中使用下面的程序代码：

```
Private Sub UserForm Click()
  Me.Caption = Range("C3").Value
End Sub
```

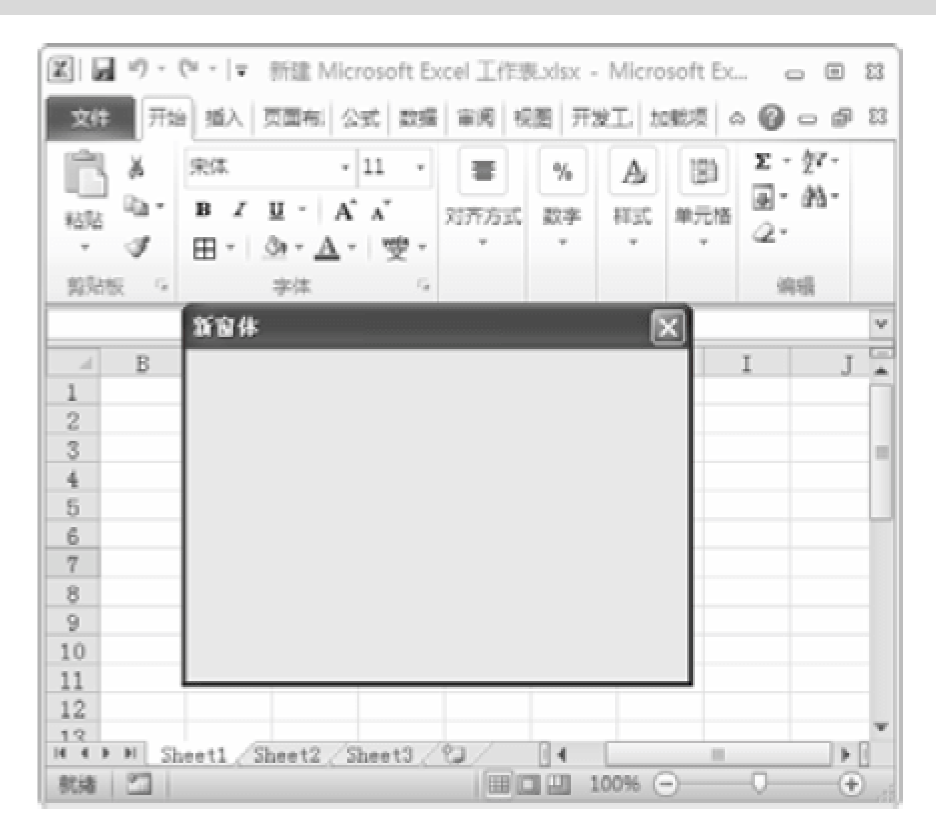


图 11-44　单击窗体改变窗体的标题栏文字

3. 窗体事件

窗体事件是对窗体操作引起程序运行的动作，如显示窗体、单击窗体和关闭窗体等动作。下面以关闭窗体为例进行讲解。

窗体关闭运行程序是通过 QueryClose 事件来实现的。

例如要实现在一个工程中，当关闭“事件 1”窗体时显示“事件 2”窗体，只需在“事件 1”窗体的代码窗口中选择 QueryClose 事件并输入以下代码：

```
Private Sub UserForm_QueryClose(Cancel As Integer,CloseMode As Integer)
   事件2.Show   ' 当“事件1”窗体关闭时，显示“事件2”窗体
End Sub
```

在该程序中可以看出，在使用 QueryClose 事件程序时，后面有两个自变量 Cancel 和 CloseMode。窗体事件程序中的自变量与工作簿和工作表事件一样，自变量的值也是从操作者操作动作取得的。

窗体关闭事件的两个自变量用法如下：

- Cancel As Integer：是否禁止关闭窗体。当值为 0 时可以关闭窗体；当为其他整数值时禁止关闭窗体。
- CloseMode As Integer：窗体的关闭模式。如果操作者是手工单击关闭按钮关闭窗体，则 CloseMode 的值为 0；如果操作者是在程序中使用 Unload 方法关闭窗体（如 Unload Me），则 CloseMode 的值为 1。

例如，禁止操作者使用窗体的关闭按钮关闭窗体，但可以使用单击窗体的方法关闭，其窗体关闭事件代码如下：

```
Private Sub UserForm_QueryClose(Cancel As Integer,CloseMode As Integer)
If CloseMode = 0 Then   ' 如果操作者使用的关闭方法是使用窗体的关闭按钮
      Cancel = 1 ' 如果上句的判断结果是单击窗体的关闭按钮，则不允许退出窗体
End If
End Sub
```

窗体单击代码如下：

```
Private Sub UserForm Click()
   Unload Me
End Sub
```

11.4.2 控件插入与使用

在介绍了窗体的插入和使用后，接下来介绍如何在窗体上插入控件以及控件的使用。

1. 认识控件工具

在窗体上插入控件必须借助控件工具箱来完成。控件工具箱是存放各种控件的工具栏。

在新插入空白窗体时会自动弹出控件工具箱，但如果控件工具箱被关闭，可以通过标准工具栏中的【工具箱】按钮来调出。

【提示】在控件工具箱中，除了【选择对象】按钮外，其他都是可以添加到窗体上的常用控件。

除了常用控件外，如果还需要添加其他控件，可以通过以下步骤把该控件添加到控件工具箱中。

01 右击控件工具箱，在弹出的快捷菜单中选择【附加控件】菜单命令，如图 11-45 所示。

02 打开【附加控件】对话框，在【可用控件】列表中选择需要添加的控件，单击【确定】按钮，如图 11-46 所示。即可将所选控件添加到工具箱中。



图 11-45　【附加控件】菜单命令



图 11-46　【附加控件】对话框

2. 从控件工具箱中插入控件

在窗体上添加控件的方法：首先在控件工具箱中选择控件，然后在窗体上拖曳出该控件，如图 11-57 所示。添加后的效果如图 11-58 所示。

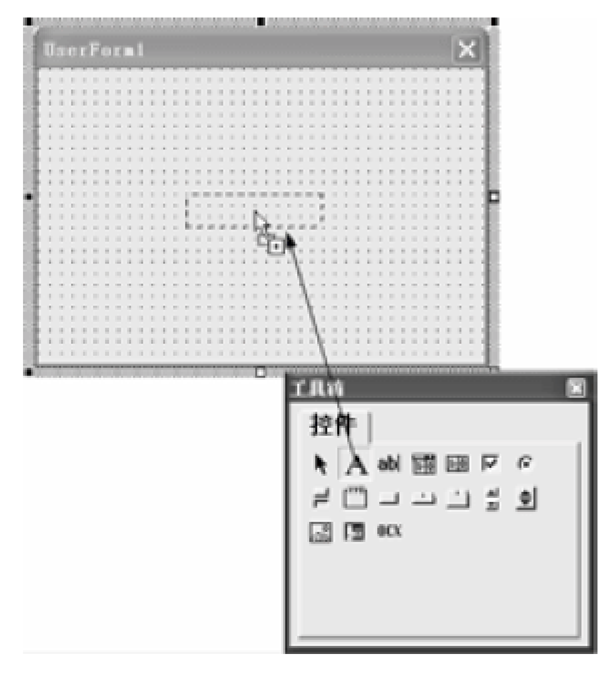


图 11-57　添加控件

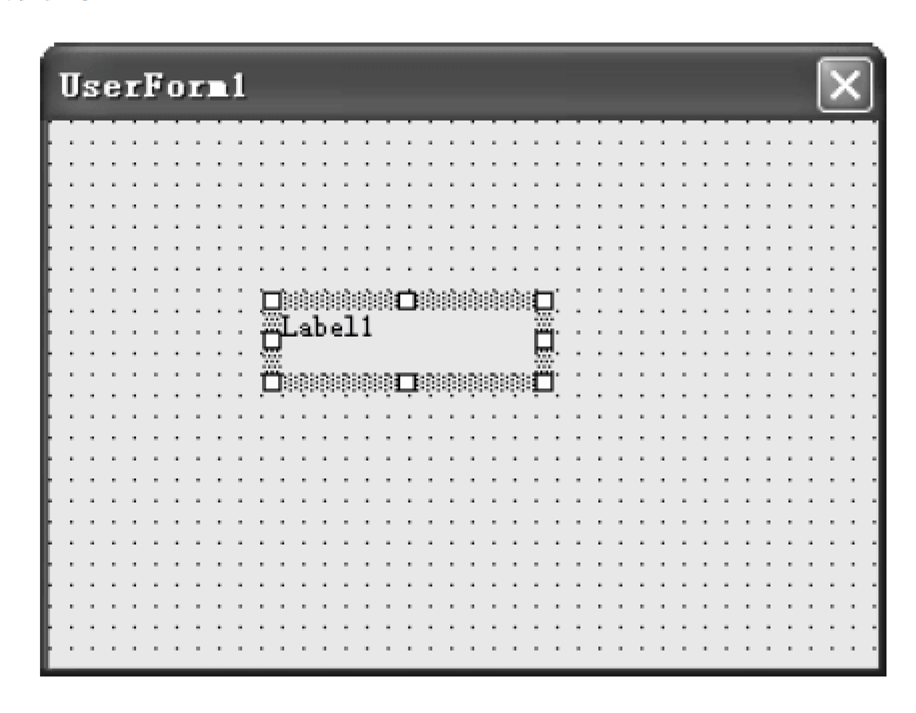


图 11-58　添加后的控件

【提示】如果要一次添加多个相同类型的控件，可以在控件工具箱中双击鼠标，然后再添加多个相同控件。



11.5 高手私房菜

技巧 1：将标签的背景设为透明

这里应用的是 Backstyle 属性。当 Backstyle 值为 0 或 fmBackStyleOpaque 时，背景透明；当 Backstyle 值为 1 或 fmBackStyleTransparent 时（默认），背景不透明。

技巧 2：如何控制控件的尺寸随内容的多少而变化

控件尺寸大小随内容的多少而改变的属性是 AutoSize。当 AutoSize 属性的值为 False 时，控件大小不随内容的多少而改变；当值为 True 时，控件的尺寸大小随内容的多少而改变。

第 12 章　窗体控件的实际应用

本章将以信息安全岗位需求调查问卷为例，详细介绍窗体控件在现实生活中的应用。通过本章的学习，可以使读者掌握岗位需求调查的问题解决方法及其设计思路。使用 Excel VBA 实现对调查问卷数据的统计分析功能，不仅能够体现 Excel VBA 高效办公的特性，而且还可以达到岗位需求调研的目的。



12.1 编制岗位需求调查问卷

岗位需求调查问卷数据管理就是在编写岗位需求调查问卷的基础上统计分析数据信息。本节主要利用【绘图】和【窗体】工具栏中的按钮，创建岗位需求调查问卷，具体内容介绍如下。

12.1.1 调查问卷的说明文字

本小节主要介绍在调查问卷中插入说明文字的方法，其中用到了【文本框】工具。使用【文本框】工具添加完说明文字后，还可以设置说明文字的大小、颜色等参数。

利用【文本框】工具编辑调查问卷的说明文字的具体操作步骤如下：

01 新建一张空白工作簿，在该工作簿的功能区选择【插入】选项卡，在【文本】选项组中选择【文本框】➢【横排文本框】菜单命令，当鼠标指针变为“↓”状时，按住鼠标左键不放并拖动至适当大小后释放，即可绘制出一个空白文本框，如图 12-1 所示。

02 在文本框内输入调查问卷的说明文字，其作用主要是说明该调查问卷的题目和目的等内容，如图 12-2 所示。

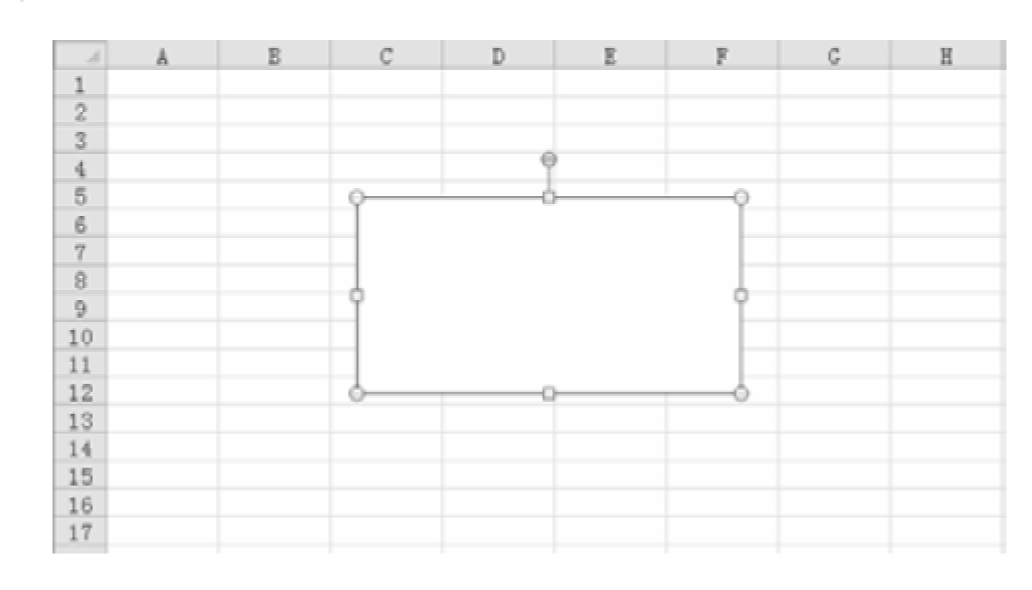


图 12-1　绘制文本框

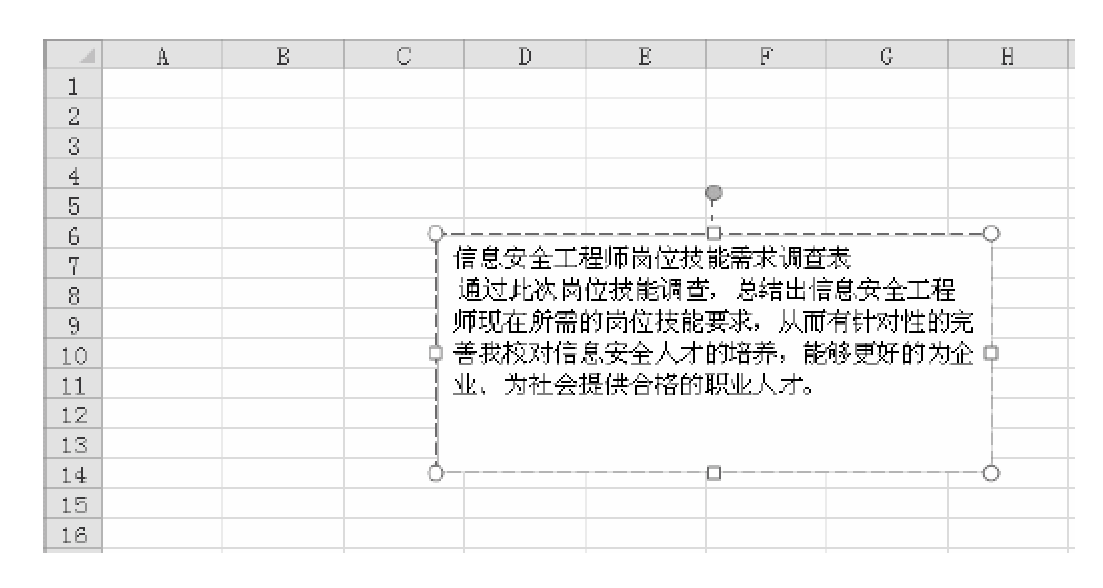


图 12-2　输入文本内容

03 选中文本框，使其处于编辑状态，然后在文本框的边框上右击，在弹出的快捷菜单中选择【设置形状格式】菜单命令，打开【设置形状格式】对话框，从中选择左侧列表框中的【线型】选项卡，在右侧的列表框中选择线型宽度为 2 磅，如图 12-3 所示。

04 选择【线条颜色】选项卡，从中选择【实线】单选按钮，设置线条颜色为深红色，并设置透明度为 0，如图 12-4 所示。



图 12-3　设置线型

图 12-4　设置线条颜色

05 选择【阴影】选项卡，在【预设】下拉列表框中选择要设置的阴影类型，然后设置其阴影颜色为浅红色，并设置其【透明度】、【大小】、【角度】、【距离】等属性，如图 12-5 所示。



图 12-5　设置文本框阴影

06 设置完毕后单击【关闭】按钮。选择文本框中的文本，设置其字体为“华文楷体”，字号为“12”，并设置其颜色为深蓝色。如果字号变大后，文本框无法容纳所有文字，可以拖动文本框的边框，使其加宽或加高。此外，还需要选择标题文本，使其居中和加粗，并右击该文本，在快捷菜单中选择【设置文字效果格式】菜单命令，弹出【设置文字效果格式】对话框，选择【阴影】选项卡，对标题设置阴影样式，修改结果如图 12-6 所示。



图 12-6　设置文本样式

12.1.2　编辑单项选择的调查内容

为了便于操作，往往会把调查问卷中涉及的问题设置成单项选择的类型，因此可以将其设置为带单选按钮的问题形式，从而方便选择，具体的操作步骤如下。

01 在功能区选择【开发工具】选项卡，在【控件】选项组中，单击【插入】下三角按钮，在弹出的控件列表中选择【表单控件】栏中的【分组框】按钮［图标］（窗体控件），在工作表中当鼠标指针变成“+”状时，按住鼠标左键不放并拖动至适当大小后释放，添加分组框 1，如图 12-7 所示。

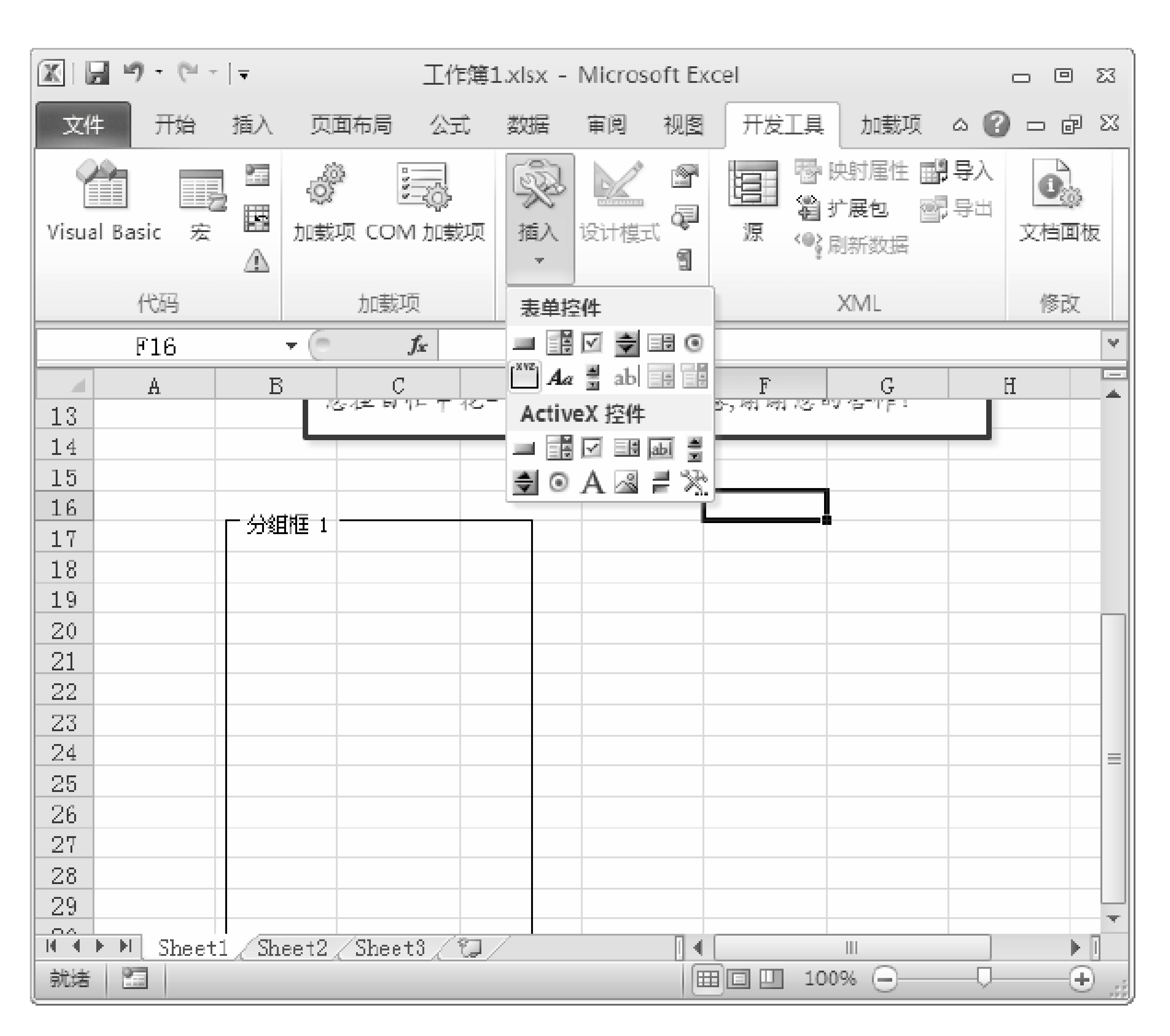


图 12-7　添加分组框

02 将新添加的分组框中的标识文字改为“性别:”，即可添加完成一个“性别”分组框，如图 12-8 所示。

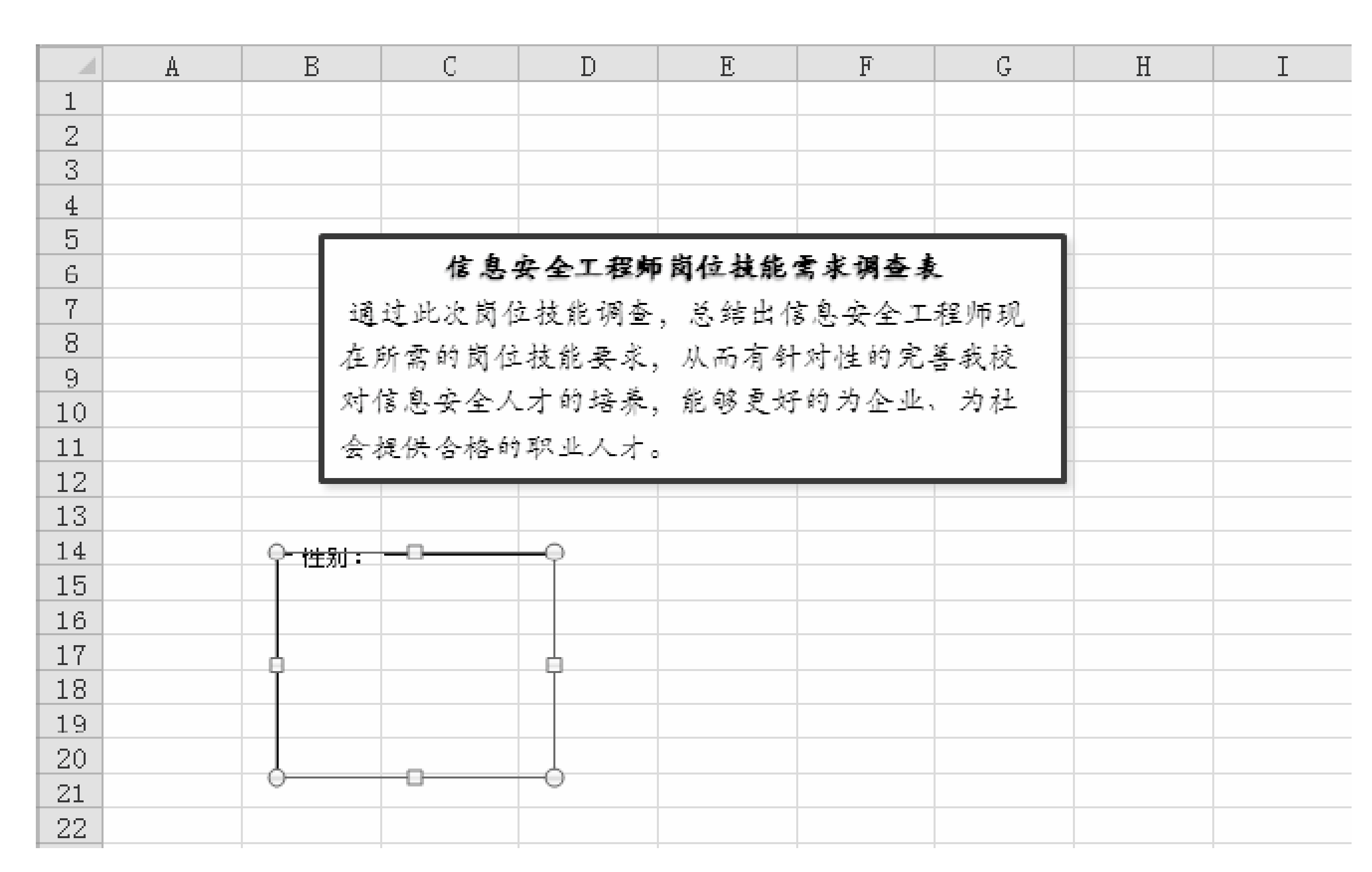


图 12-8　修改分组框标识文字

03 在【开发工具】选项卡的【控件】选项组中，单击【插入】下三角按钮，在弹出的控件列表中选择【选项按钮】按钮 ◉ （窗体控件），在分组框内拖动至适当大小后释放鼠标左键，即可添加一个选项按钮，如图 12-9 所示。

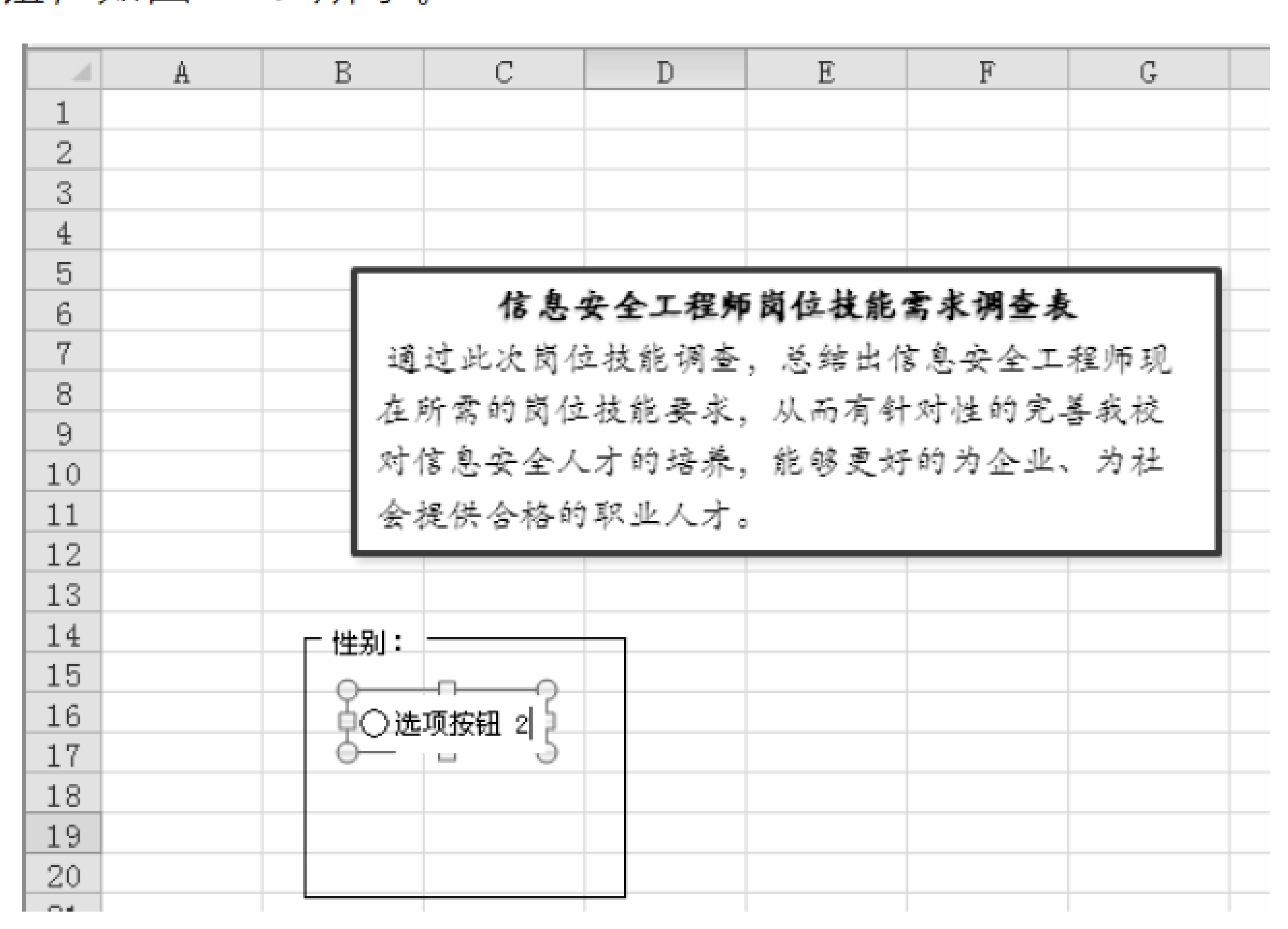


图 12-9　添加选项按钮

04 选中添加的选项按钮，使其处于可编辑状态，直接修改其中的文字；或者右击该选项按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令进行修改。

05 将选项按钮的标识文字修改为“男”；按同样的方法，在分组框内添加另一个选项按钮，并将其标识文字修改为“女”，如图 12-10 所示。

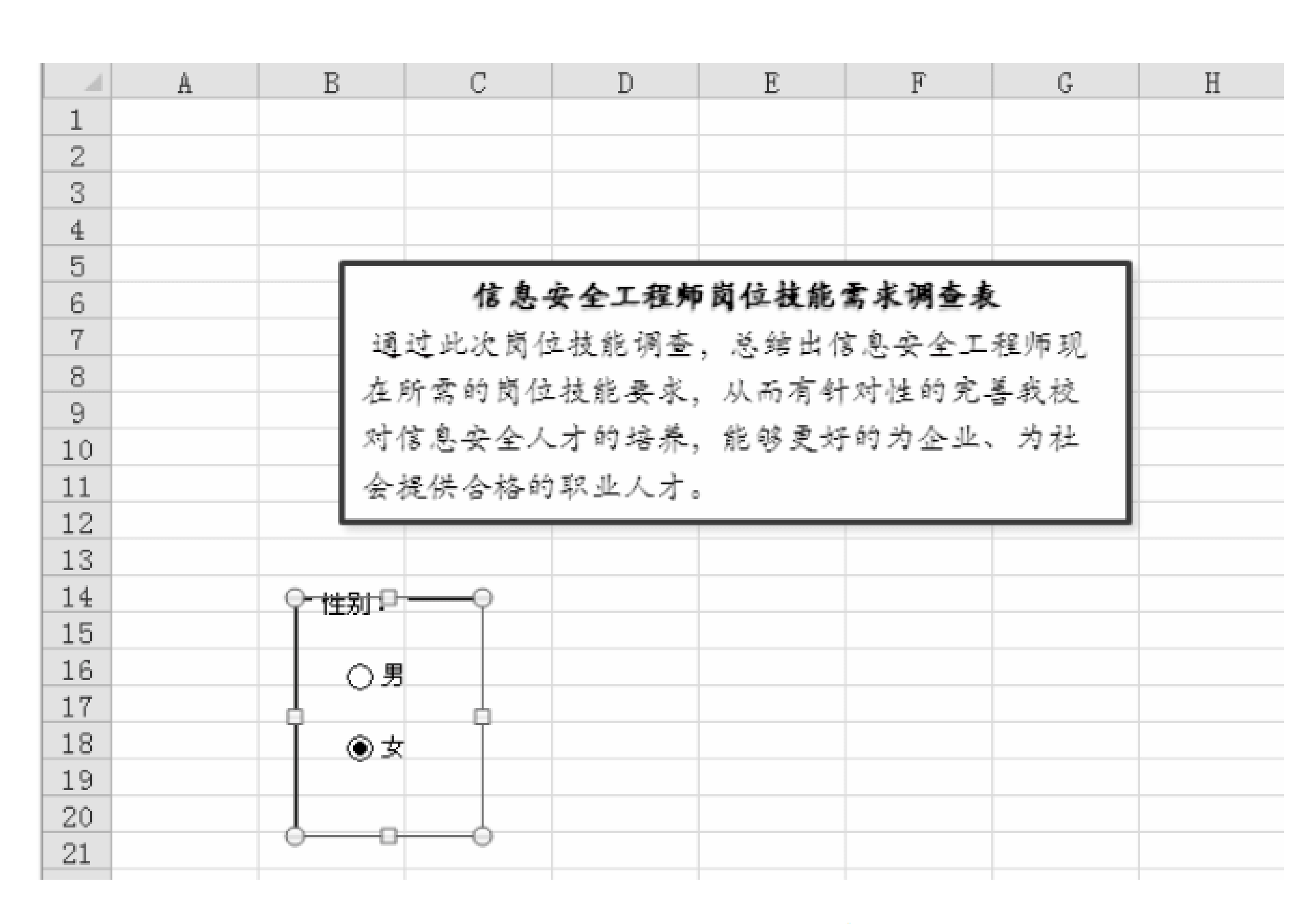


图 12-10　添加选项按钮并修改文字

06 为了使调查问卷的选项按钮更加醒目，可以为其设置一些特殊效果。右击添加的【男】选项按钮，在弹出的快捷菜单中选择【设置控件格式】菜单命令，打开【设置控件格式】对话框，如图 12-11 所示。



图 12-11　【设置控件格式】对话框

07 选择【颜色与线条】选项卡，在【填充】设置区中的【颜色】下拉列表中选择【填充效果】选项，打开【填充效果】对话框，在【渐变】选项卡中选择【预设】单选按钮，从【预设颜色】下拉列表中选择【雨后初晴】选项，然后在【底纹样式】设置区中选中【中心辐射】单选按钮，并在【变形】设置区中选择第 2 种样式，如图 12-12 所示。





图 12-12 【颜色与线条】选项卡

08 设置完成后，单击【确定】按钮返回【设置控件格式】对话框，继续单击【确定】按钮返回工作表中，设置后的效果如图 12-13 所示。



图 12-13 设置选项按钮的样式效果

09 按照同样的方法将添加的【女】选项按钮设置为同样的填充效果。

10 按照上述方法，分别添加【年龄】、【学历】和【职业】分组框，并分别设置其相应的选项内容，得到的最终效果如图 12-14 所示。

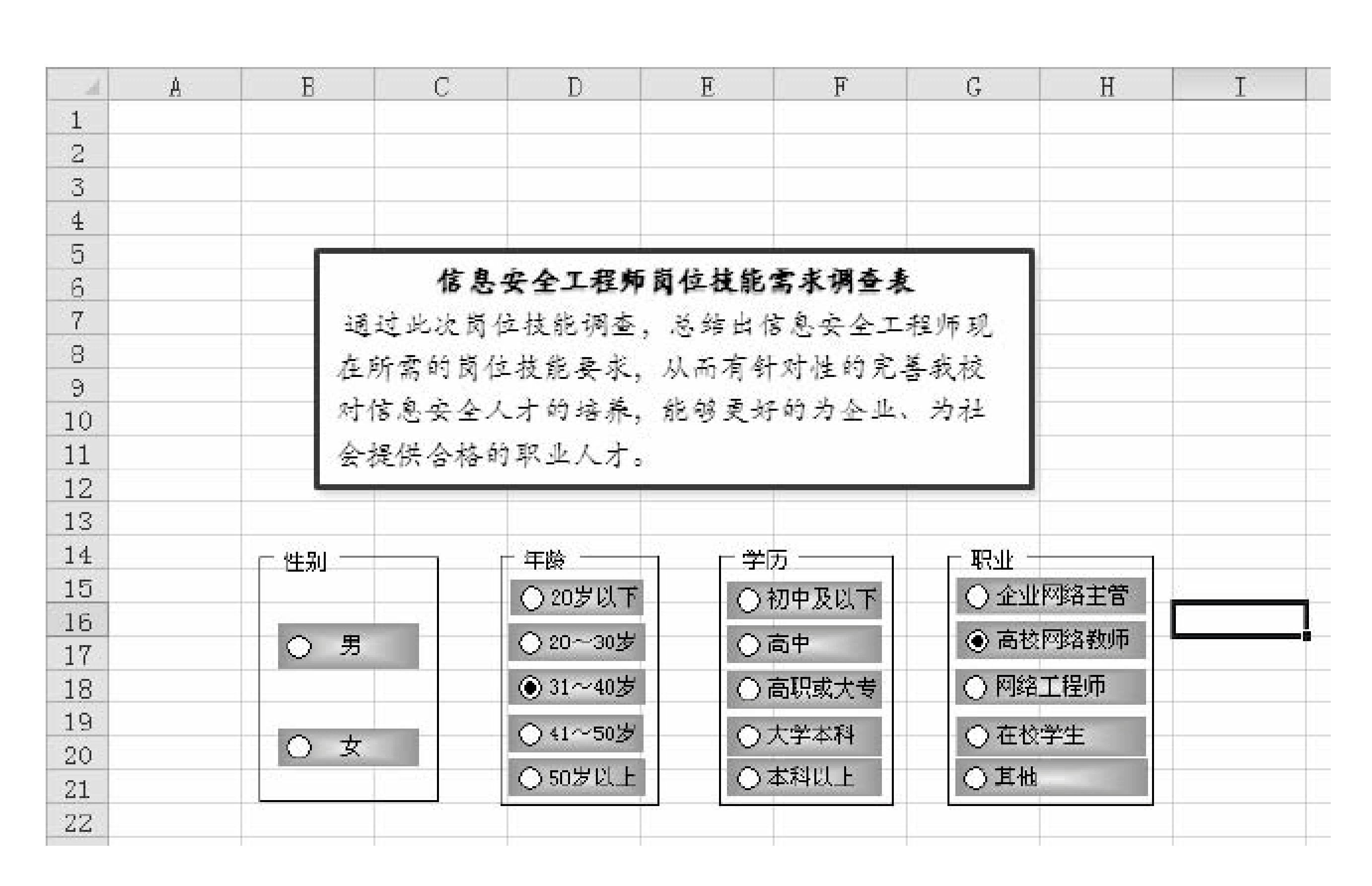


图 12-14　设置其他分组框内容

添加的选项按钮比较多，需要将其排列整齐。可以通过右击选项按钮的方式将其选中，并按住鼠标左键不放进行拖动即可。对于不需要的选项按钮，可以按 Delete 键直接删除。另外，如果添加的选项按钮的大小不合适，可以利用鼠标在边框的 8 个控点上进行拖动，即可实现选项按钮在垂直方向、水平方向和斜向的大小变化。

12.1.3　编辑下拉列表的调查内容

在岗位需求调查问卷中，很多问题都会包括多种选项内容，而且文字比较多，如果全都是用单选按钮的方式来实现，会占用比较多的文本空间，所以有必要将其设计成下拉列表的形式，具体的操作步骤如下。

01 在功能区【开发工具】选项卡的【控件】选项组中，单击【插入】下三角按钮，在弹出的控件列表中选择【标签】按钮 Aa（窗体控件），待鼠标指针变成“+”状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，即可添加一个标签，如图 12-15 所示。

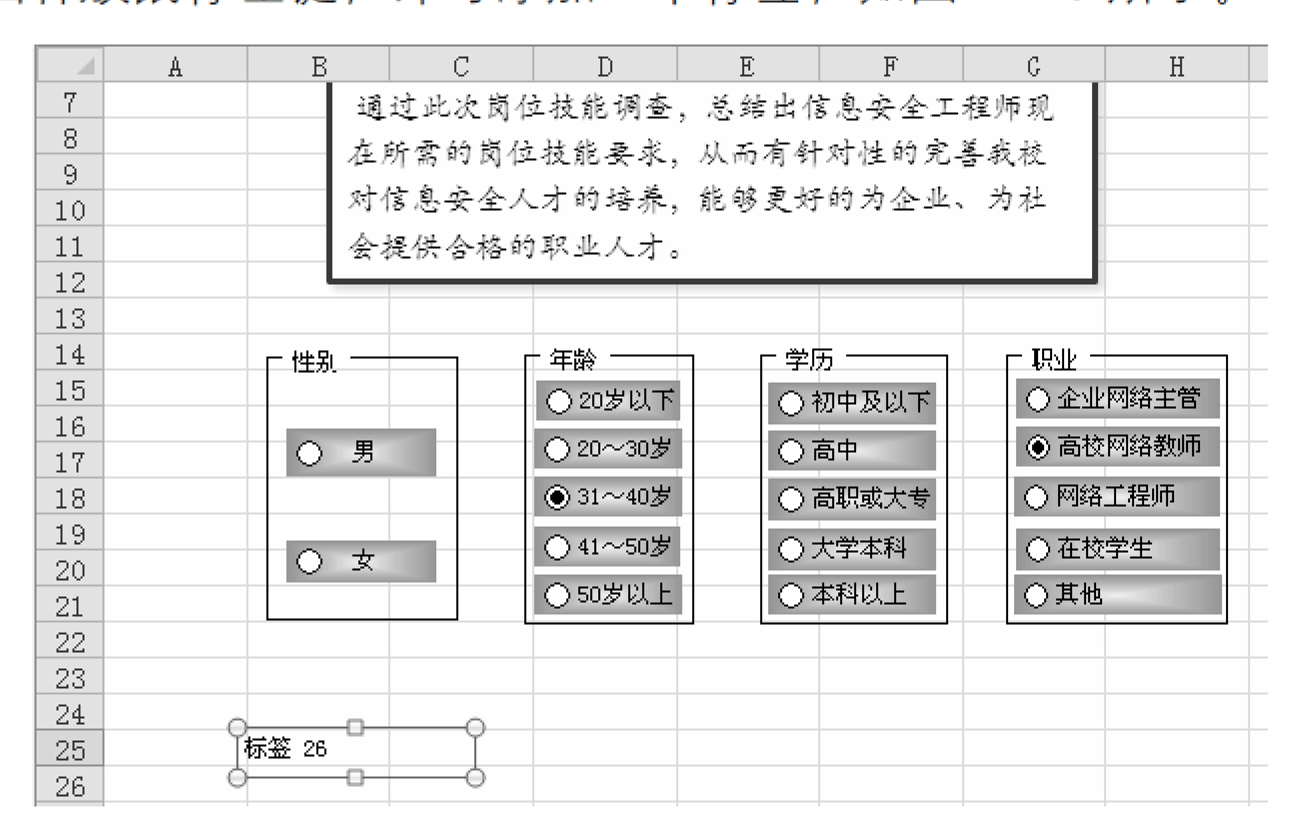


图 12-15　添加标签控件

02 将添加的标签标识文字修改为“1、贵公司的业务主要包含哪些方面？”，并对标签的大小和位置进行相应的调整，如图 12-16 所示。



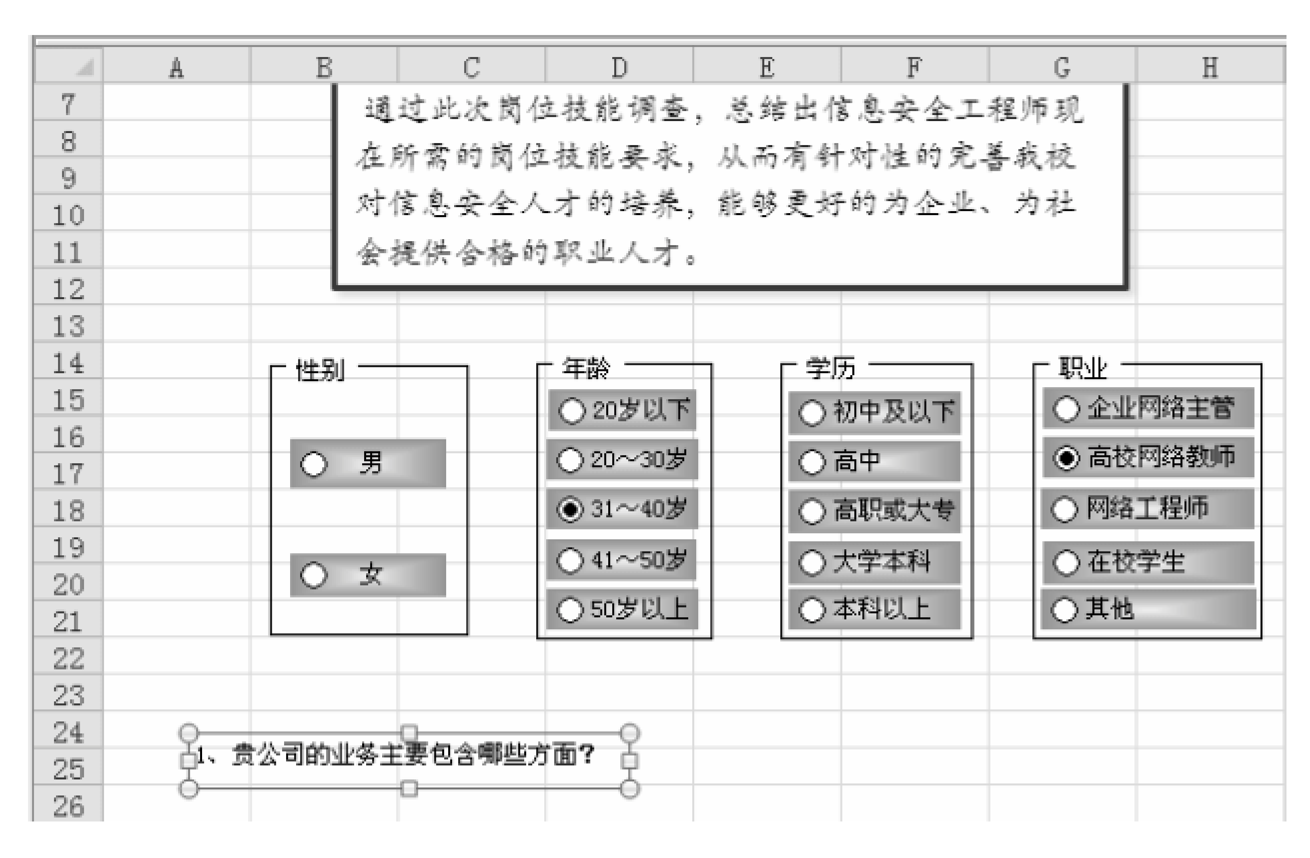


图 12-16　修改标签标识

03 在功能区【开发工具】选项卡的【控件】选项组中，单击【插入】下三角按钮，在弹出的控件列表中选择【组合框】按钮，待鼠标指针变成"+"状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，即可添加一个组合框，如图 12-17 所示。

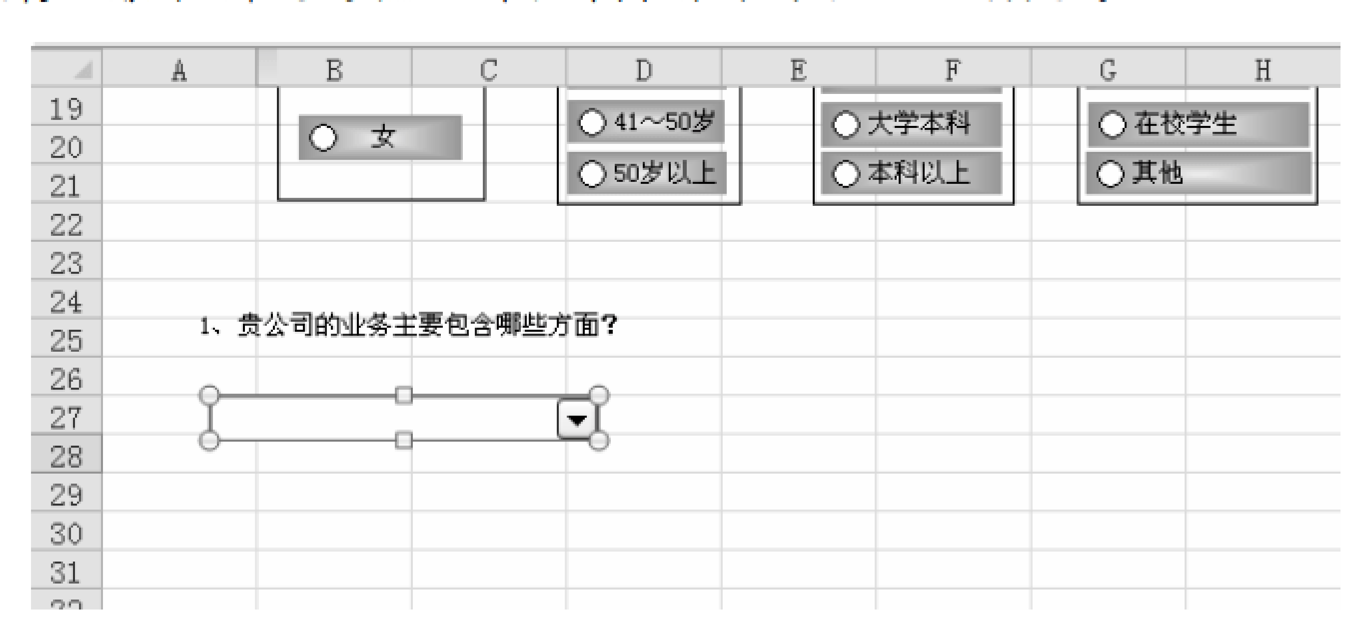


图 12-17 添加组合框控件

04 右击添加的组合框，在弹出的快捷菜单中选择【设置控件格式】菜单命令，即可打开【设置控件格式】对话框，如图 12-18 所示。



图 12-18　【设置控件格式】对话框

05 选择【控制】选项卡，单击【数据源区域】文本框右侧的【折叠】按钮，可将该对话框折叠起来，单击问卷内容工作表标签 Sheet2，切换到问卷内容工作表中，选择第一题待选答案所对应的单元格区域"A2：A6"，此时在【设置控件格式】对话框中即可显示添加的数据源区域，如图 12-19 所示。

	A	B	C
1	1、贵公司的业务主要包含哪些方面?	4、贵公司在工程中都做哪些服务器?	8、贵公司历年中的数据库服务
2	A 系统集成	A Web	A Sql Server 2000
3	B 网络安全	B FTP服务器	B Sql Server 2005
4	C 安防监控	C 数据库服务器	C Oracle 9i
5	D 产品代理	D Exchange邮件服务器、MD邮件服务器等	D Oracle 10g
6	E 其他	E Lotus	E DB2
7	2、贵公司都安排有哪些技术岗位?	F OA服务器	F SyBase
8	A 系统工程师	[illegible]	[illegible]
9	B 数据库工程师		
10	C 网站前台工程师	建都基于哪些环境?	邮件服 些邮件服务器产品?
11	D 网站后台开发工程师	A Windows Server 2000+IIS+ASP.NET	A Exchange 2003
12	E 售后工程师	B Windows Server 2003+IIS+ASP.NET	B Exchange 2007
13	F 网络工程师	C Windows Server 2000+APACHE+PHP	C MD
14	G 信息安全工程师	D Windows Server 2003+APACHE+PHP	D Lotus
15	H 网站开发、管理工程师	E Windows Server 2000+IIS+JSP	E 其他
16	I 安防监控工程师	F Windows Server 2003+IIS+JSP	10、贵公司在项目中曾使用到过 品?
17	J 其他	G ____linux +APACHE+PHP	A 梭子鱼
	3、您认为上述技术岗位上的工程师近两年应该掌握的操作系统是哪	H ____linux+APACHE+JSP	B 其他

设置控件格式

Sheet2!$A$2:$A$6

Sheet1 Sheet2 Sheet3

图 12-19　设置数据源区域

【提示】问卷内容工作表 Sheet2 的数据必须提前录入。

06 单击【设置控件格式】对话框中的【展开】按钮，随即展开该对话框，为美化添加的组合框，选中【三维阴影】复选框，如图 12-20 所示。



图 12-20　展开【设置控件格式】对话框



07 单击【确定】按钮返回工作表 Sheet1 中，当前组合框还处于编辑模式，单击该工作表的任意空白单元格，退出编辑模式。然后单击组合框右侧的下三角按钮▼，会弹出第一题的待选答案列表，如图 12-21 所示。

08 按照前面设置第一题的方法来添加其他问题。在实际操作的过程中，可以右击添加的标签或组合框，在弹出的快捷菜单中选择【复制】菜单命令，在合适的位置进行粘贴，并修改其文本内容等，这样可以快速完成其他问题及答案的添加，添加完成后最终效果如图 12-22 所示。

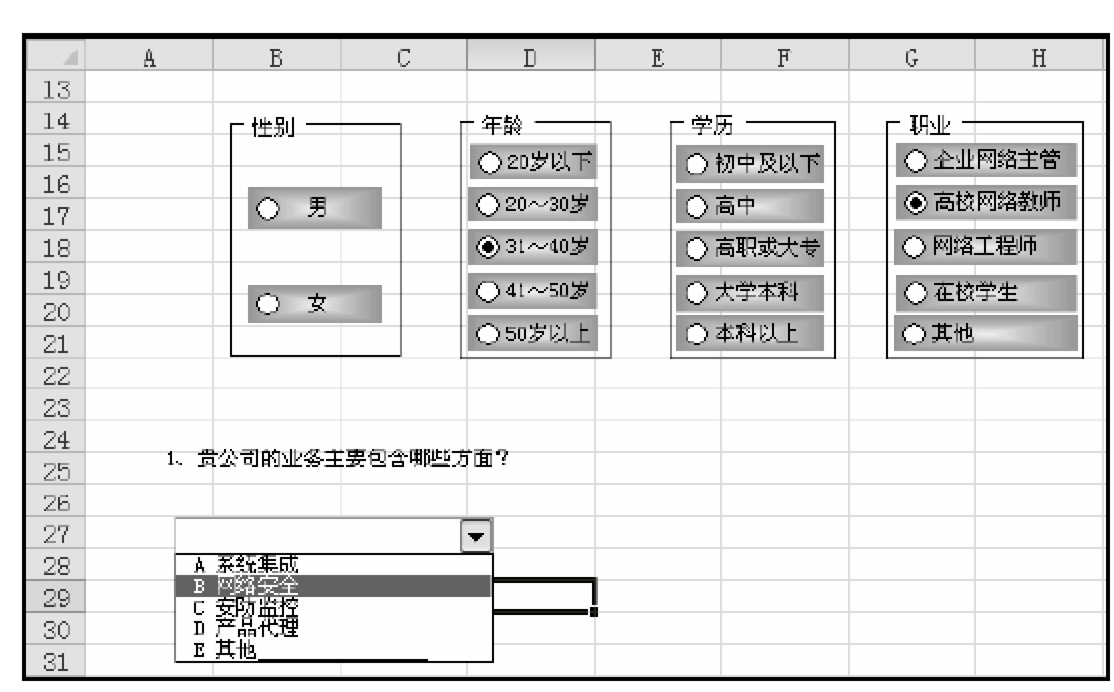


图 12-21　显示第一题的待选答案



图 12-22　设置其他标签和组合框

12.1.4 保护工作表

添加好的岗位需求调查问卷内容，在进行调查使用时，是不允许内容被修改的，所以需要将已经设置好的选项内容保护起来，这主要通过对调查问卷内容所在的工作表设置保护，具体的操作步骤如下。

01 在工作表 Sheet1 中，选择功能区的【审阅】选项卡，在【更改】选项组中，单击【保护工作表】按钮，打开【保护工作表】对话框。在该对话框中选中【保护工作表及锁定的单元格内容】复选框，在【取消工作表保护时使用的密码】文本框中输入想要设置的密码，例如〝111111〞，然后在【允许此工作表的所有用户进行】列表中撤销选中【选定锁定单元格】和【选定未锁定的单元格】复选框，如图 12-23 所示

02 单击【确定】按钮，打开【确认密码】对话框，在【重新输入密码】文本框中再次输入设置的密码〝111111〞，如图 12-24 所示。



图 12-23　【保护工作表】对话框

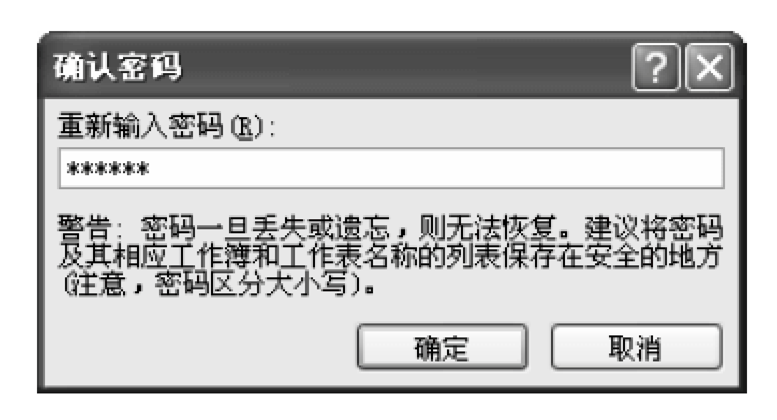


图 12-24　【确认密码】对话框

03 单击【确定】按钮返回工作表，此时工作表已经受保护了，如果再对工作表进行编辑或修改等操作，系统都会自动弹出一个提示信息框。

同样，为了防止更改表中的数据，按照前面的方法可将问卷内容工作表进行保护设置。并且为了使整体效果更加美观，还可以将问卷内容工作表进行隐藏，具体操作方法如下。

在该工作表的功能区选择【开始】选项卡，在【单元格】选项组中选择【格式】➤【隐藏和取消隐藏】➤【隐藏工作表】菜单命令，即可将其隐藏，如图 12-25 所示。



图 12-25　隐藏工作表

至此，已经完成了关于信息安全工程师岗位需求调查问卷的设计和编辑过程。用户只需将搜集到的调查数据按照相应的问题选项录制在该调查问卷中即可。

如果要取消对工作表的隐藏，则在【开始】选项卡的【单元格】选项组中单击【格式】➤【隐藏和取消隐藏】➤【取消隐藏工作表】菜单命令，即可取消对工作表的隐藏。



12.2 汇总岗位需求调查问卷结果

只是创建了岗位需求调查问卷的内容样式，并不能满足获得岗位需求调查结果的目的，还需要对调查结果进行统计分析。下面将为控件设置相应的单元格链接，并利用 Excel VBA 程序将调查问卷的结果汇总成记录信息。

12.2.1 为控件创建单元格链接

通过为控件设置单元格链接，可以实现将调查结果的选项内容与某一单元格或单元格区域联系起来，从而将每一个调查问题的选择结果转化为相应的数字信息，并以数字形式保存在该单元格或单元格区域中，从而获得最终汇总调查结果。

为控件创建单元格链接的具体操作步骤如下。

01 在工作表中创建用来暂时存放各选项按钮和组合框信息的单元格区域，其形式如图 12-26 所示。

图 12-26 创建存放各选项信息的单元格区域

02 为分组框中的选项按钮创建单元格链接。右击【性别】分组框中的【男】单选按钮，在弹出的快捷菜单中选择【设置控件格式】菜单命令，打开【设置控件格式】对话框，如图 12-27 所示。



图 12-27　【设置控件格式】对话框

03 在【控制】选项卡中单击【单元格链接】文本框右侧的【折叠】按钮，在“调查问卷结果汇总”单元格区域中选择用于存放“性别”数据的单元格 A44，如图 12-28 所示。

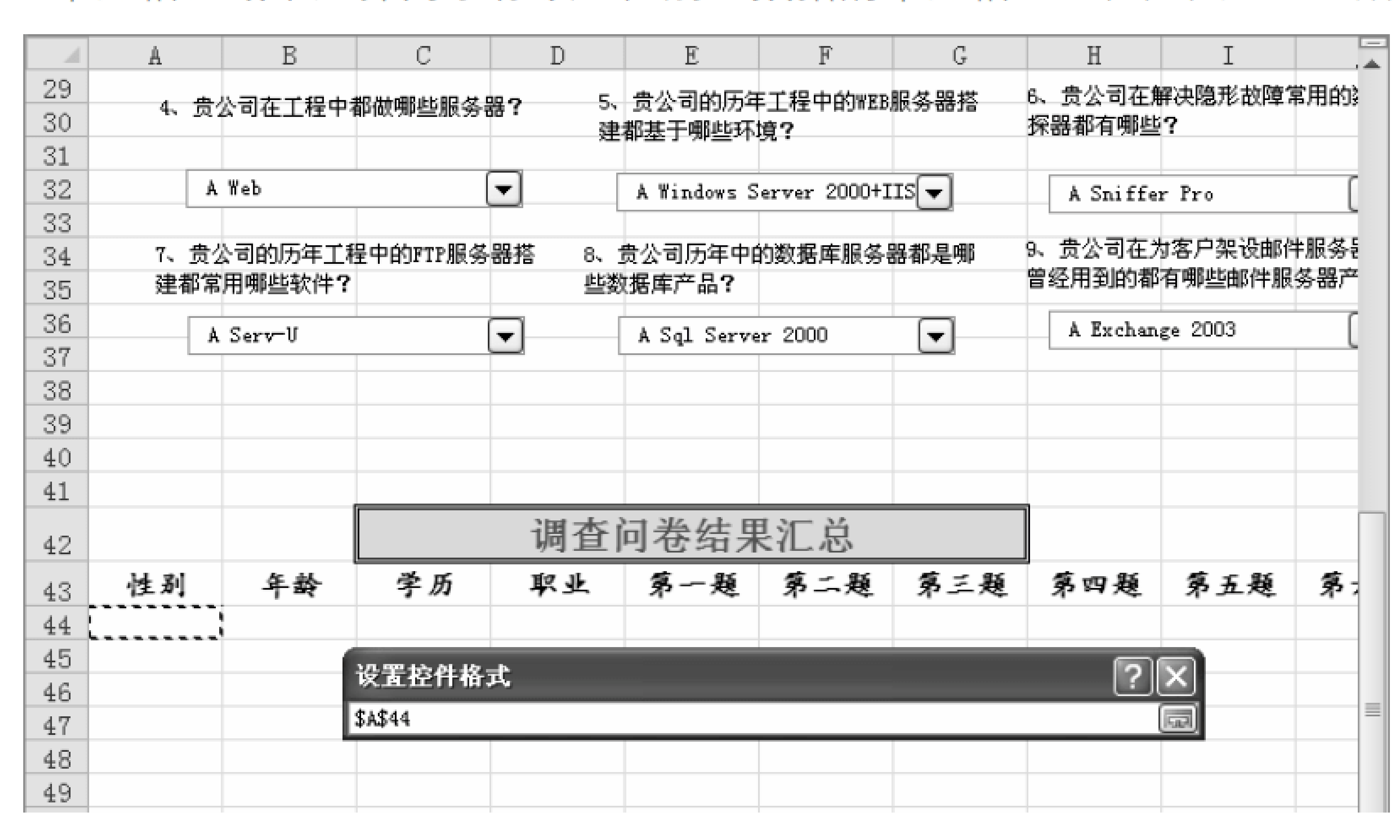


图 12-28　设置性别单元格的链接

04 单击【设置控件格式】对话框中的【展开】按钮，展开【设置控件格式】对话框，单击【确定】按钮即可。

05 此时如果单击【性别】分组框中的【男】单选按钮，在单元格 A44 中将自动显示与之对应的数值“1”；如果单击【女】单选按钮，则会自动显示数值为“2”，如图 12-29 所示。





图 12-29　选择性别时显示相应数值

【提示】汇总数据中显示的数据是当前选项所选选项的排序号。例如，选择【年龄】选项中的第三个【31~40岁】，在对应的汇总结果单元格中就会显示“3”。

06 按照同样的方法，分别为其他组合框中的选项按钮创建单元格链接。接下来为组合框创建单元格链接。右击第一题的组合框，在弹出的快捷菜单中选择【设置控件格式】菜单命令，打开【设置控件格式】对话框，在【控制】选项卡中单击【单元格链接】文本框右侧的【折叠】按钮，在"调查问卷结果汇总"单元格区域中选择用于存放"第一题"数据的单元格 E44，如图 12-30 所示。



图 12-30　设置“第一题”的组合框链接

07 单击【设置控件格式】对话框中的【展开】按钮，返回到【设置控件格式】对话框，单击【确定】按钮，完成设置。

08 按照同样的方法，为其他组合框设置单元格链接。当在调查问卷中进行选择时，即可将

其对应的选项转化为相应的数值显示在“调查问卷结果汇总”单元格区域中，如图 12-31 所示。

公司的业务主要包含哪些方面？　安防监控
2、贵公司都安排有哪些技术岗位？　F 网络工程师
3、您认为上述技术岗位上的工程师近两年应该掌握的操作系统是哪些？　B Windows Server 2003
公司在工程中都做哪些服务器？　Lotus
5、贵公司的历年工程中的WEB服务器搭建都基于哪些环境？　E Windows Server 2000+IIS
6、贵公司在解决隐形故障常用的数据嗅探器都有哪些？　B Ethereal
司的历年工程中的FTP服务器搭用哪些软件？　VSFtpd
8、贵公司历年中的数据库服务器都是哪些数据库产品？　C Oracle 9i
9、贵公司在为客户架设邮件服务器时，曾经用到的都有哪些邮件服务器产品？　B Exchange 2007

调查问卷结果汇总

年龄	学历	职业	第一题	第二题	第三题	第四题	第五题	第六题
2	3	3	3	6	2	5	5	2

Sheet1　Sheet2　Sheet3

图 12-31　显示调查问卷中的结果

12.2.2　实现自动记录功能

要想实现调查结果的自动记录功能，首先要确定保存调查结果信息的工作表，然后编写具有自动记录功能的 VBA 程序代码，具体的操作步骤如下：

01 右击工作表标签 Sheet2，在弹出的快捷菜单中选择【重命名】菜单命令，将工作表 Sheet2 命名为“问卷内容”，使用同样的方法将工作表 Sheet3 命名为“数据汇总”，用来确定保存调查结果信息的工作表，然后将工作表 Sheet1 中“调查问卷结果汇总”单元格区域中的数据整理到该工作表中，如图 12-32 所示。



图 12-32　设置单元格区域的数据显示

02 选择功能区【开发工具】选项卡，单击【代码】选项组中的【Visual Basic】按钮，打开 VBA 代码窗口，在【工程管理器】窗口的任意位置右击，从弹出的快捷菜单中选择【插入】➢【模块】菜单命令，插入一个新模块，双击该模块，在菜单栏中选择【插入】➢【过程】菜单命令，打开【添加过程】对话框，在【名称】文本框中输入“自动记录”，在【类型】选项组中选中【子

程序】单选按钮，在【范围】选项组中选择【公共的】单选按钮，如图 12-33 所示。

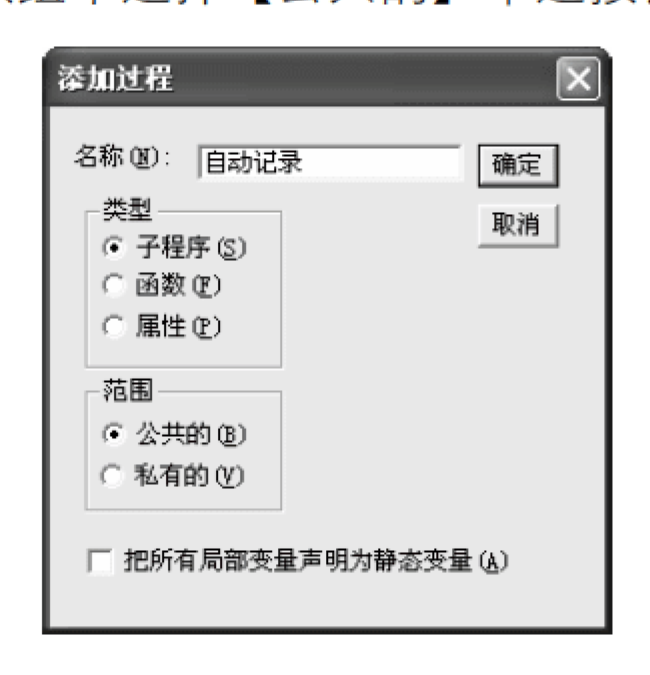


图 12-33　【添加过程】对话框

03 单击【确定】按钮系统将自动添加一个"自动记录"公有子过程，然后在该子过程中添加以下程序代码。

```
Public Sub 自动记录()
Dim i As Integer '定义用来保存工作表中占用单元格行数的整型变量
Dim k As Integer  '定义用来保存调查结果记录行数的整型变量
   i=Sheets("数据汇总").[A1].CurrentRegion.Rows.Count'工作表"数据汇总"的记录行数
   k=i - 3     '工作表中有 3 行单元格保存字段名
   Sheets("Sheet1").Select
   Range("A44:N44").Select  '选择需要保存数据的区域
   Selection.Copy  '对选择的数据区域进行复制
   Sheets("数据汇总").Activate  '激活工作表"数据汇总"
   Cells(i + 1, 2).Select '选择工作表中最后一条记录的下一行单元格区域
   ActiveSheet.Paste '粘贴数据
   Cells(i + 1, 1).Value = k + 1 '实现数据记录的自动编号功能
   Sheets("Sheet1").Select
   Application.CutCopyMode = False  '释放进行复制操作的单元格区域
   MsgBox "记录已成功保存，谢谢您的合作！", vbOKOnly, "提交信息"
End Sub
```

在这段代码中，首先定义两个分别用来记录工作表"数据汇总"中单元格的记录行数和现有调查结果的记录行数的整型变量 i 和 k。然后使用 Copy 方法对工作表 Sheet1 中选定的单元格区域"A44:N44"进行复制操作，再使用 Paste 方法将这些数据粘贴到工作表"数据汇总"中最后一条记录的下一行单元格中，并实现数据记录的自动编号功能。最后释放工作表 Sheet1 中进行复制操作的单元格区域，并以消息对话框的形式输出记录的保存结果。

04 代码输入完毕后，为便于快速执行该代码，可在工作表中为其设置 VBA 程序运行按钮。即选择功能区【开发工具】选项卡，单击【控件】选项组中的【插入】按钮，在弹出的下拉列表中单击【按钮】控件▭，然后在工作表 Sheet1 的适当位置按住鼠标左键不放并拖动至适当大小后释放，此时系统自动打开【指定宏】对话框，在【宏名】列表框中选择【自动记录】选项，如图 12-34 所示。

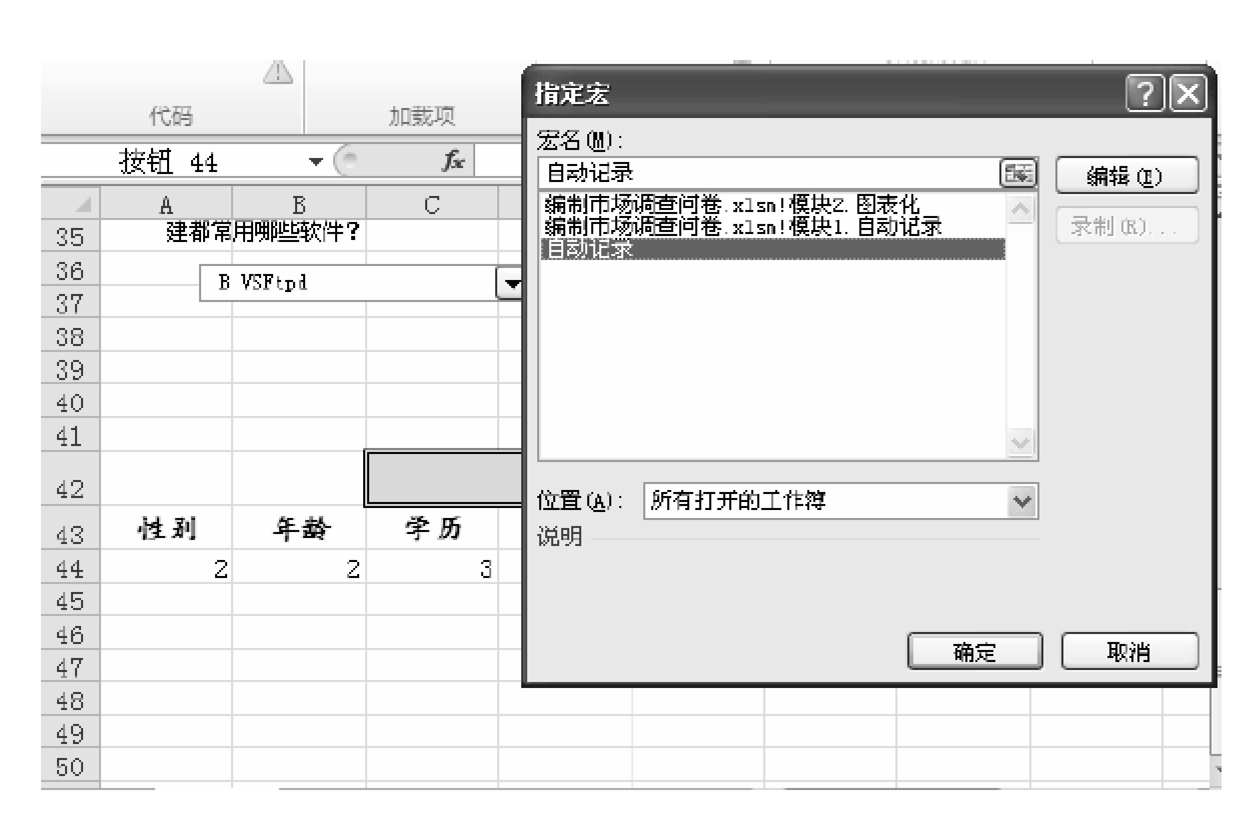


图 12-34　指定宏名称

05 单击【确定】按钮，返回到工作表 Sheet1 中，在该按钮上单击鼠标右键，从弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮的标志文字更改为“提交问卷结果”，如图 12-35 所示。



图 12-35　修改按钮文字

06 右击添加的“提交问卷结果”按钮，在弹出的快捷菜单中选择【设置控件格式】菜单命令，打开【设置控件格式】对话框，在【字体】列表框中选择“华文新魏”，在【字形】列表框中选择“常规”，在【字号】列表框中选择“14”，在【下划线】下拉列表中选择“单下划线”选项，在【颜色】下拉列表中选择“深蓝色”，如图 12-36 所示。



图 12-36　【设置控件格式】对话框

07 设置完毕后，单击【确定】按钮，按钮文字的设置效果如图 12-37 所示。

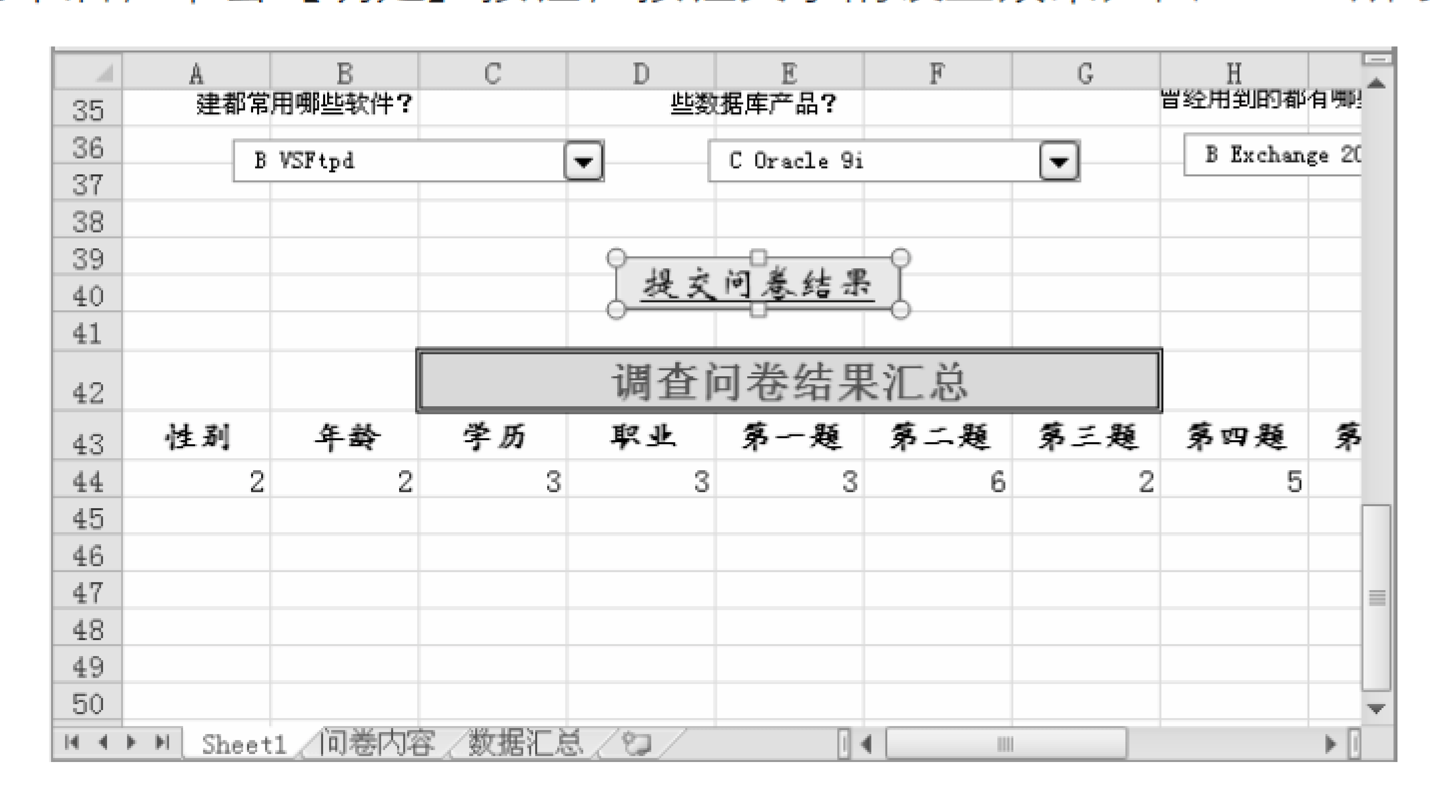


图 12-37 按钮文字的设置效果

08 单击工作表的任意位置，可退出该按钮的编辑状态。此时当完成问卷内容后，将鼠标移至【提交问卷结果】按钮，指针将变为☝，单击该按钮即可对本次结果进行提交，并弹出提示信息，如图 12-38 所示。



图 12-38 提交信息

09 完成所有问卷结果的提交后，可以切换到【数据汇总】工作表中，调查问卷结果的保存情况如图 12-39 所示。



调查问卷结果汇总

序号	基本信息				问题编号								
	性别	年龄	学历	职业	一	二	三	四	五	六	七	八	九
1	2	2	3	3	3	6	2	5	5	2	2	3	2
2	2	2	3	3	3	6	2	5	5	2	2	3	2
3	2	2	3	3	3	6	2	5	5	2	2	3	2

图 12-39 调查结果显示



12.3 统计分析岗位需求调查问卷结果

前面所创建的调查问卷，除了具有汇总调查结果的功能，还具有自动记录的功能，但是还不能对调查结果进行统计分析。下面将使用 Excel VBA 对自动记录的调查问卷数据进行统计分析，从而实现需求调研的目的。

12.3.1 创建岗位需求调查数据统计表

在利用 VBA 实现岗位需求调查问卷结果的统计和分析之前，首先需要创建一张用于保存统计结果的调查数据统计表，即在该表中实现对各个选项结果的分类统计，具体的操作步骤如下。

01 在“数据汇总”工作表的后面插入一张新表，将其命名为“调查统计表”，并在该工作表中添加相应字段内容，其中“选项编号”列用于显示调查问卷中各个答案对应的序列号，如图 12-40 所示。

	A	B	C	D	E	F	G	H	I	J	K
1	岗位需求调查数据统计表										
2	选项编号	基本信息				题目序号					
3		性别	年龄	学历	职业	一	二	三	四	五	六
4	1										
5	2										
6	3										
7	4										
8	5										

Sheet1　问卷内容　数据汇总　调查统计表

图 12-40　插入新表并添加字段内容

02 为便于看表中各列所对应的“选项编号”的意义，可为其添加上相应的批注。例如，右击“性别”字段名称，在弹出的快捷菜单中选择【插入批注】菜单命令，在弹出的【插入批注】文本框中输入批注的内容，如图 12-41 所示。

03 按照同样的方法，为其他字段插入相应的批注。然后当鼠标指针移到插入批注的字段右上角的红色“斜三角”处，系统将自动显示出批注的内容。

【提示】用户如果需要修改或删除插入的批注，可以右击插入批注的单元格，在弹出的快捷菜单中选择【编辑批注】、【删除批注】或【显示/隐藏批注】菜单命令。

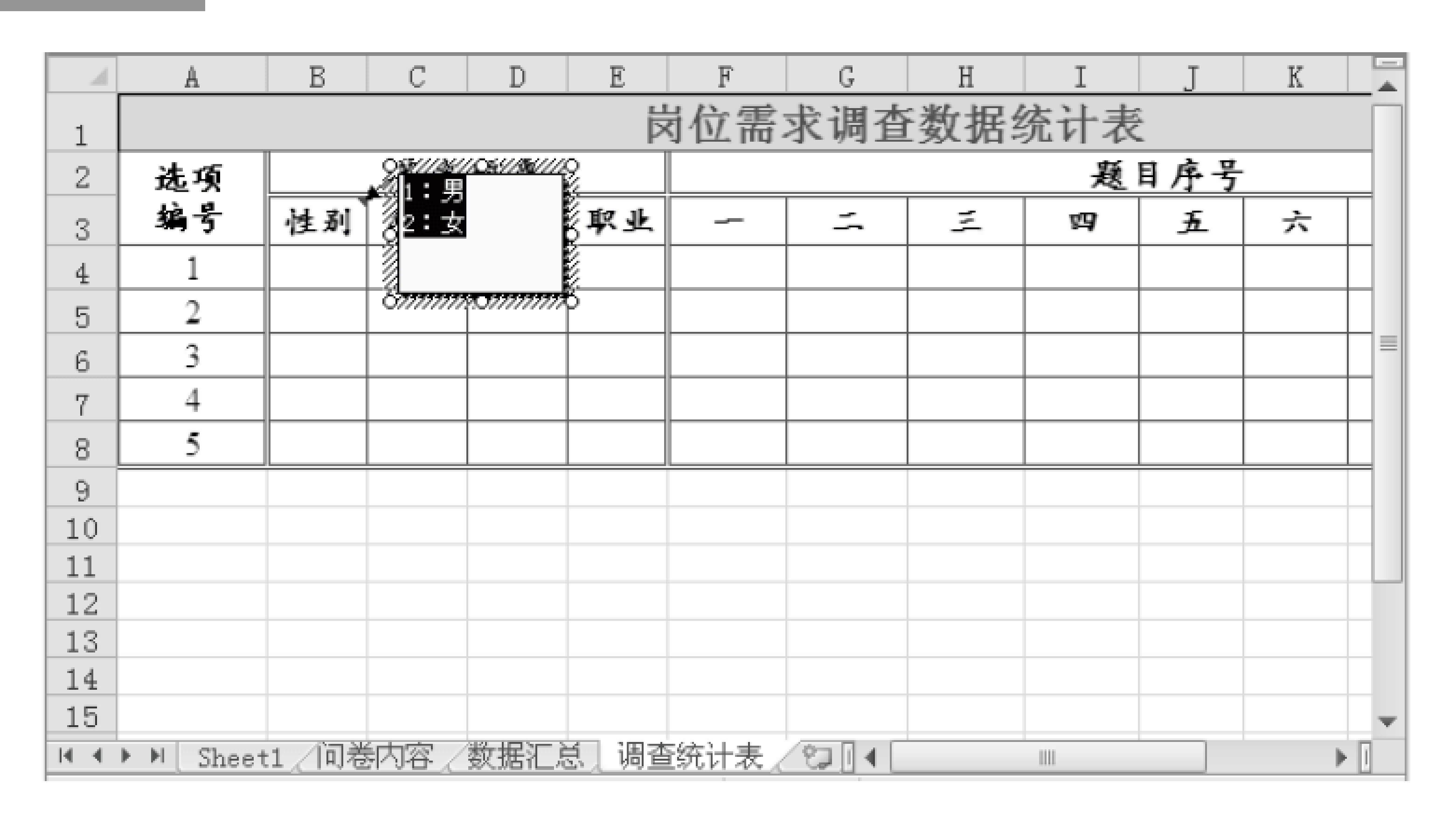


图 12-41 插入批注

12.3.2 统计分析调查问卷结果

市场调查数据统计表创建完成后，需要利用 Excel VBA 程序代码对调查问卷的结果进行统计分析，并将统计结果图表化，以便清晰直观地查看调查结果。

1. 自动统计调查结果

利用 VBA 代码对汇总的数据进行统计分析，具体的操作步骤如下。

01 选中【调查统计表】工作表，选择功能区【开发工具】选项卡，在【控件】选项组中单击【插入】按钮，在弹出的列表中单击【ActiveX 控件】组中的【命令按钮】按钮 (ActiveX 控件)，在工作表的合适位置添加用于执行 VBA 程序的控件按钮，如图 12-42 所示。

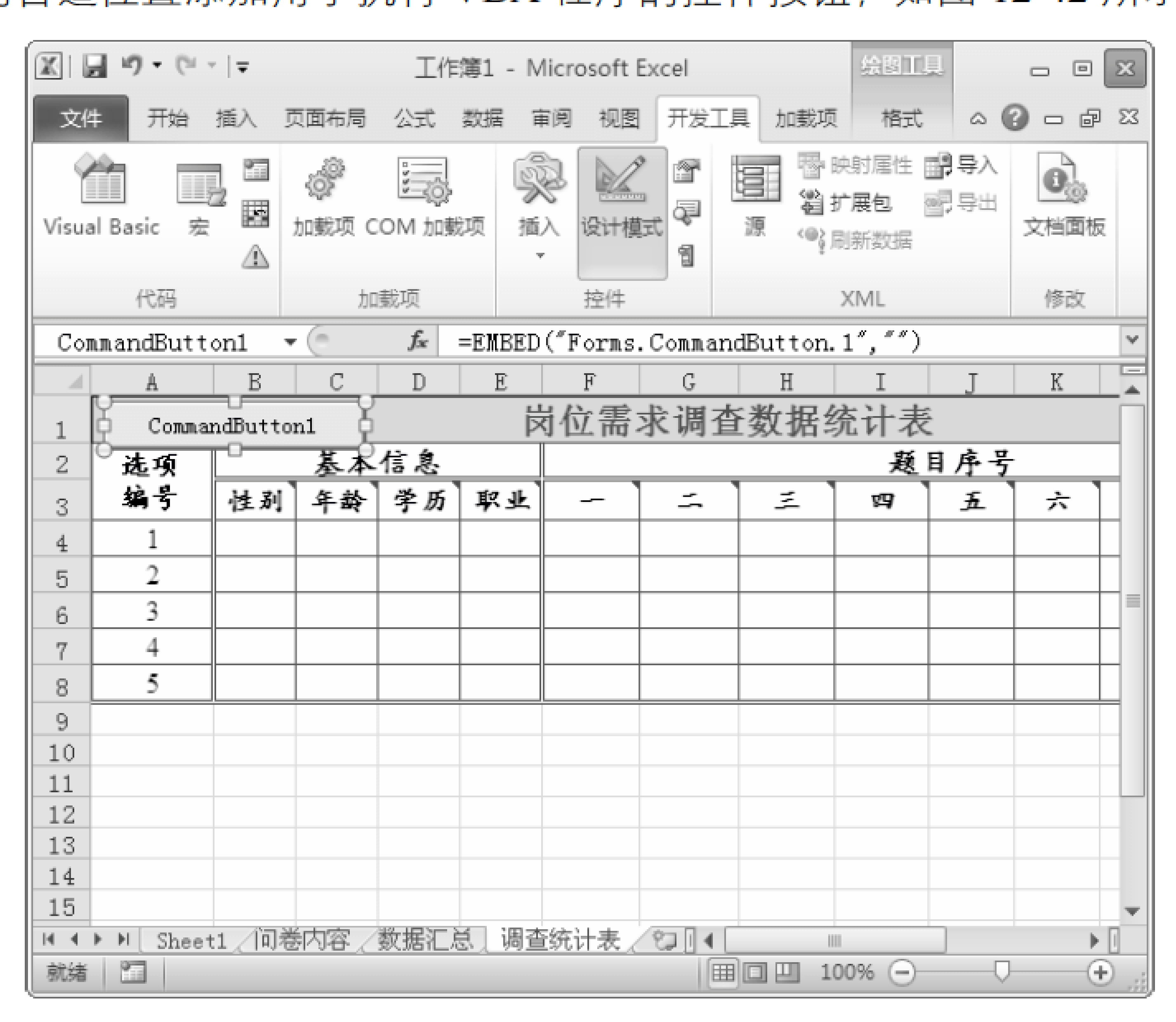


图 12-42 添加命令按钮控件

02 右击该按钮，在弹出的快捷菜单中选择【属性】菜单命令，打开该按钮的【属性】窗口，在【按字母序】选项卡的【名称】文本框和 Caption 文本框中均输入"统计数据"，如图 12-43 所示。



图 12-43　修改按钮属性

03 双击新添加的按钮，即可打开 VBA 代码窗口，在其中输入以下程序代码。

```
    Private Sub 统计数据_Click()
    Dim i, j As Integer    '定义用于循环的整型变量
    Dim count As Integer   '定义用于保存记录数的整型变量
    Dim temp As Variant    '定义用于读取单元格内容的变体型变量
      count = Sheets("数据汇总").[A1].CurrentRegion.Rows.count '统计工作表"数据汇总"
'中的记录数
      Sheets("调查统计表").Select
      For Each temp In Range("B4:N8")
         temp.Value = ""   '将单元格区域"B4:N8"中的值置空
      Next temp
      For i = 4 To count
         For j = 2 To 15
            Select Case Worksheets("数据汇总").Cells(i, j)
    ' 对工作表"数据汇总"中的单元格内容执行 Select Case 语句
            Case 1
            Worksheets("调查统计表").Cells(4, j) = Worksheets("调查统计表").Cells(4,
j) + 1
    ' 如果"数据汇总"中单元格值为 1，则"调查统计表"中对应的单元格的统计数值加 1
            Case 2
              Worksheets("调查统计表").Cells(5, j) = Worksheets("调查统计表").Cells(5,
j) + 1
    ' 如果"数据汇总"中单元格值为 2，则"调查统计表"中对应的单元格的统计数值加 1
            Case 3
            Worksheets("调查统计表").Cells(6, j) = Worksheets("调查统计表").Cells(6,
j) + 1
    ' 如果"数据汇总"中单元格值为 3，则"调查统计表"中对应的单元格的统计数值加 1
```

```
            Case 4
             Worksheets("调查统计表").Cells(7, j) = Worksheets("调查统计表").Cells(7,
j) + 1
    ' 如果“数据汇总”中单元格值为 4，则“调查统计表”中对应的单元格的统计数值加 1
            Case 5
             Worksheets("调查统计表").Cells(8, j) = Worksheets("调查统计表").Cells(8,
j) + 1
    ' 如果“数据汇总”中单元格值为 5，则“调查统计表”中对应的单元格的统计数值加 1
            End Select
        Next j
      Next i
      Sheets("调查统计表").Select
    End Sub
```

在这段代码中，首先利用 For Each 循环语句将工作表【调查统计表】中的单元格区域“B4:N8”置空，然后在双重 For 循环中使用 Select Case 语句自动统计调查问卷数据结果，最后返回到工作表“调查统计表”中。

04 程序输入完毕后，在【调查统计表】工作表中单击功能区【开发工具】选项卡中的【控件】选项组中的【设计模式】按钮“”，退出该命令按钮的编辑状态。

05 单击【统计数据】按钮即可实现对调查问卷结果的统计，根据图中的显示结果，可以了解到在回答该调查问卷的 6 个人中，有 3 个人是男性、3 个人是女性，如图 12-44 所示。

统计数据	岗位需求调查数据统计表									
选项编号	基本信息				题目序号					
	性别	年龄	学历	职业	一	二	三	四	五	六
1	3	1	2							3
2	3	3		1	2		3	3		3
3		2	4	3	4				3	
4				2		2				
5								3	3	

图 12-44　统计结果显示

2. 统计结果图表化

为了能够更加形象直观地显示出调查问卷的统计结果，可以利用 VBA 代码为其创建图表，使统计的结果图表化，具体操作步骤如下。

01 在 VBA 代码窗口中的【工程资源管理器】窗口的任意位置右击，在弹出的快捷菜单中选择【插入】➤【模块】菜单命令，然后双击该模块，并在该模块中输入如下程序代码。

```
    Sub 图表化()
    Dim cht As Chart     '定义一个图表类型变量 cht
       If Charts.count > 0 Then '判断是否存在已创建的图表，如果存在则先将其删除
```

```
        Charts.Delete
      End If
      Set cht = Charts.Add  '将添加的图表赋值给 cht
       Charts(1).Name = "数据分析图表" '命名图表所在的工作表
      With cht  '使用 With 语句简化相同的操作
         .Location Where:=xlLocationAsNewSheet   '在新工作表中添加图表
         .ChartType = xlColumnClustered   '选择图表类型为“簇状柱形图”
         .SetSourceData Source:=Sheets("调查统计表").Range("A2:N8"), PlotBy:=xlRows
   ' 选择数据来源
         .SeriesCollection(1).Name = "=""选项编号 1"""   '设置图表系列的名称
         .SeriesCollection(2).Name = "=""选项编号 2"""
         .SeriesCollection(3).Name = "=""选项编号 3"""
         .SeriesCollection(4).Name = "=""选项编号 4"""
         .SeriesCollection(5).Name = "=""选项编号 5"""
         .HasTitle = True  '设置图表显示标题
         .ChartTitle.Characters.Text = "调查问卷数据分析" '设置图表标题
         .Axes(xlCategory, xlPrimary).HasTitle = True  '设置图表显示 x 轴坐标标题
         .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "调查内容"
   ' 设置 x 轴坐标标题
         .Axes(xlValue, xlPrimary).HasTitle = True '设置图表显示 y 轴坐标标题
         .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "选择数目"
   ' 设置 y 轴坐标标题
      End With
   End Sub
```

在这段代码中，首先利用 If 语句判断是否存在图表，如果存在，则先将其删除，这样可以保证在多次执行该程序时，只保留当前最新创建的图表。然后使用 Set 方法创建一个图表，在 With 语句中设置图表的各项内容，包括指定位置、设置图表类型、选择数据来源和设置图表标题等。该程序代码的作用是为调查问卷的选项结果创建柱形图图表，并将其显示在新建的工作表“数据分析图表”中。用户可根据此图表分析调查问卷的选择情况。

02 为简化操作，在【调查统计表】工作表中为上述程序设置启动方式。在【插入】选项卡中，单击【剪贴画】按钮，在打开的剪贴画列表框中选择要插入的剪贴画，如图 12-45 所示。



图 12-45　插入剪贴画



03 调节插入图形的大小，将其置于工作表的合适位置。然后右击该剪贴画，在弹出的快捷菜单中选择【指定宏】菜单命令，即可打开【指定宏】对话框，在【宏名】列表框中选择【图表化】选项，如图12-46所示。



图12-46 【指定宏】对话框

04 单击【确定】按钮，返回工作表中，单击任意位置退出剪贴画的编辑状态。此时将鼠标移动到剪贴画上，待鼠标指针变为时单击，即可运行指定的程序，在新工作表"数据分析图表"中将统计的结果图表化，如图12-47所示。

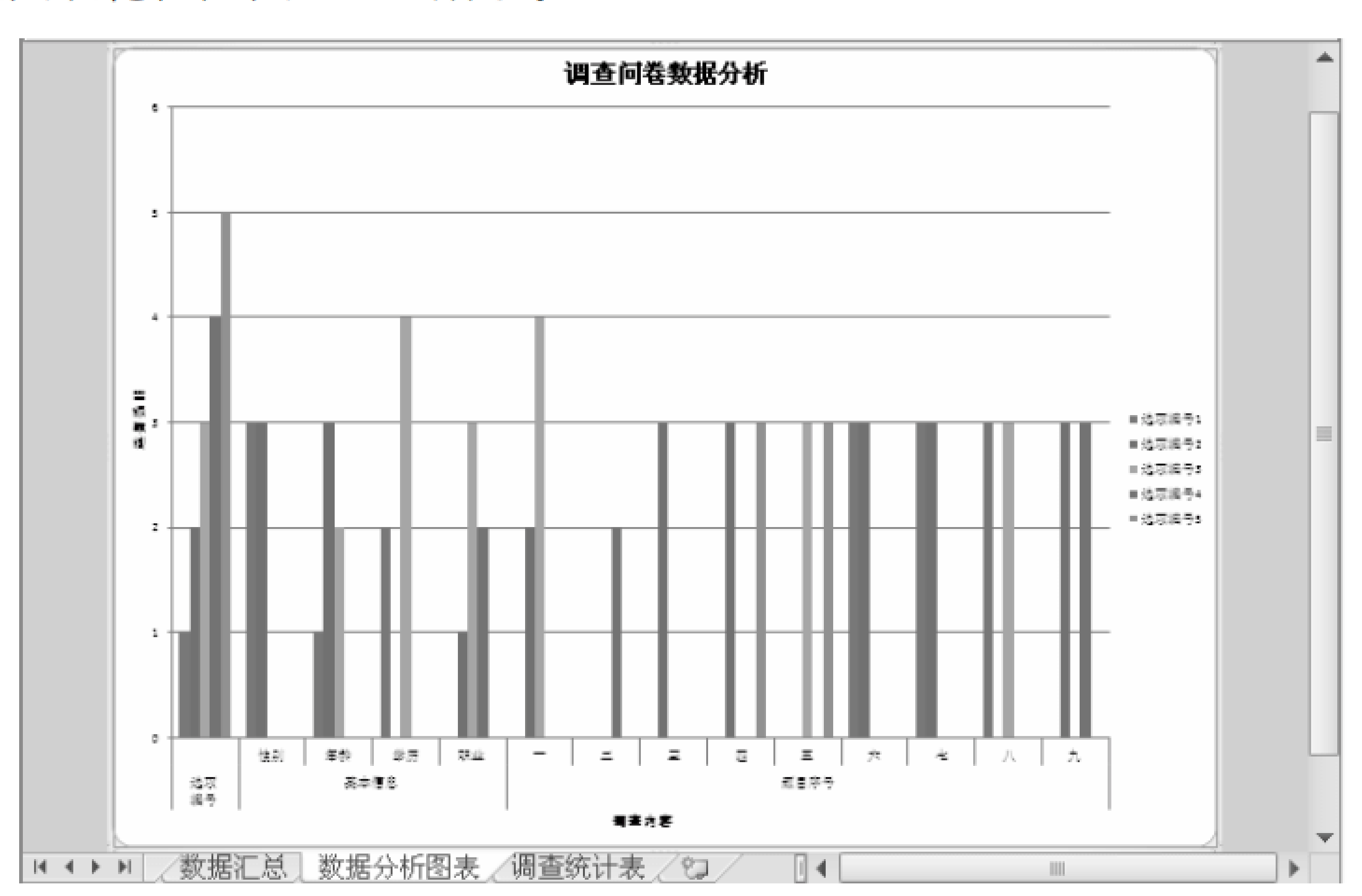


图12-47 将统计的结果图表化

05 当对"调查统计表"工作表中的数据进行更新后，再次单击用于创建图表的按钮时，系统会先将之前创建的图表删除，并弹出如图12-48所示的提示信息框，单击【删除】按钮，即可创建新的图表。



图 12-48　提示信息框

到此，已经完成了关于岗位需求调查问卷的制作及分析过程。用户可在实际工作中根据具体的情况进行相应的设置。



12.4 高手私房菜

技巧 1：在输入文字后调整文本框的位置和大小

如果文本框的位置和大小不合适，可以将鼠标移至文本框的边界上对其进行调整。将鼠标置于边框的 8 个控点上拖动，即可实现文本框在垂直方向、水平方向和斜向的大小变化；而将鼠标置于边框的其他位置拖动，即可实现文本框位置的移动。

技巧 2：调整窗体运行时控件的可见性

若窗体运行时不需要显示该控件，可以通过设置控件的 Visible 属性来实现。

例如，在窗体中插入两个文本框控件，然后在窗体的代码窗口中输入如下代码：

```
Private Sub UserForm_Initialize()
   TextBox2.Visible = False  '隐藏文本框 2
End Sub
```

输入完毕后，运行该窗体后，文本框 1 显示，文本框 2 不显示。

Visible 属性只有在窗体运行状态下才会发生作用，在 VBA 代码编辑器中的编辑状态下无法发生效力。

第 13 章　自定义函数的实际应用

Excel 2010 中提供了丰富的内置函数，有三百多种。利用这些内置函数可以快捷、准确地完成大量的数据处理任务，大大地提高了办公效率。除此之外，还可以通过在【加载宏】对话框中加载 Analysis ToolPak（分析工具库）得到更多的函数。通过以上内置函数并不一定能够满足所有的需求，这时可以通过自定义函数来解决问题。下面将详细介绍自定义函数的相关内容。



13.1 认识自定义函数

要想实现 Excel 2010 自定义函数，首先要对 VBA 基础知识有所了解，因为自定义函数是利用 Excel 2010 VBA 编写的函数。

13.1.1 使用自定义函数

自定义函数可以像内置函数一样正常使用，具体操作方法介绍如下。

【例 13.1】编写自定义函数，使其能够获取当前工作簿中工作表的个数，具体操作步骤如下。

01 新建或打开一个 Excel 2010 工作簿，按 Alt+F11 组合键打开 VBA 代码窗口，在【工程资源管理器】中的任意位置右击，在弹出的快捷菜单中选择【插入】➤【模块】菜单命令，即可插入一个模块，双击该模块，在其中输入以下函数代码，如图 13-1 所示。

```
Function 工作表个数()      '自定义函数为“工作表个数”
   工作表个数 = ThisWorkbook.Sheets.Count    '为自定义函数指定执行代码
End Function
```

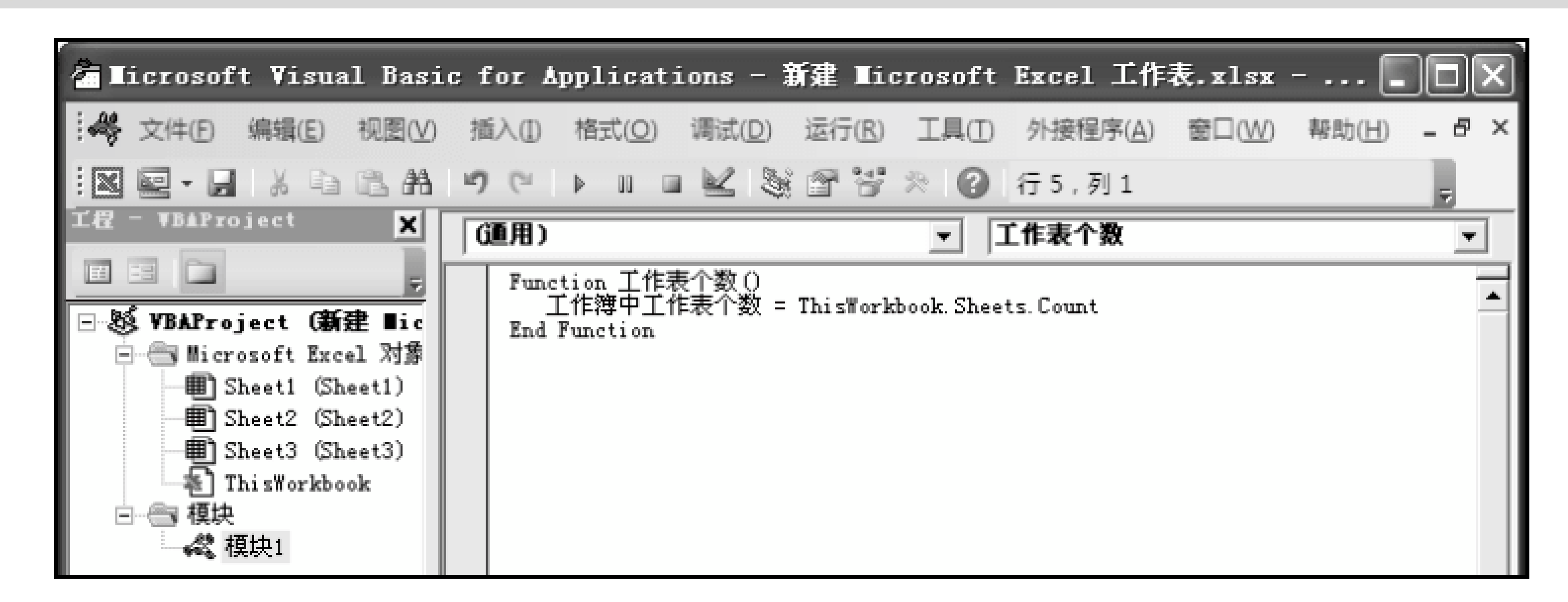


图 13-1　添加模块并输入代码

02 输入完毕后，返回工作表中，使用定义的函数设置公式“=工作表个数（）”，就可以得到当前工作簿中工作表的个数，如图 13-2 所示。

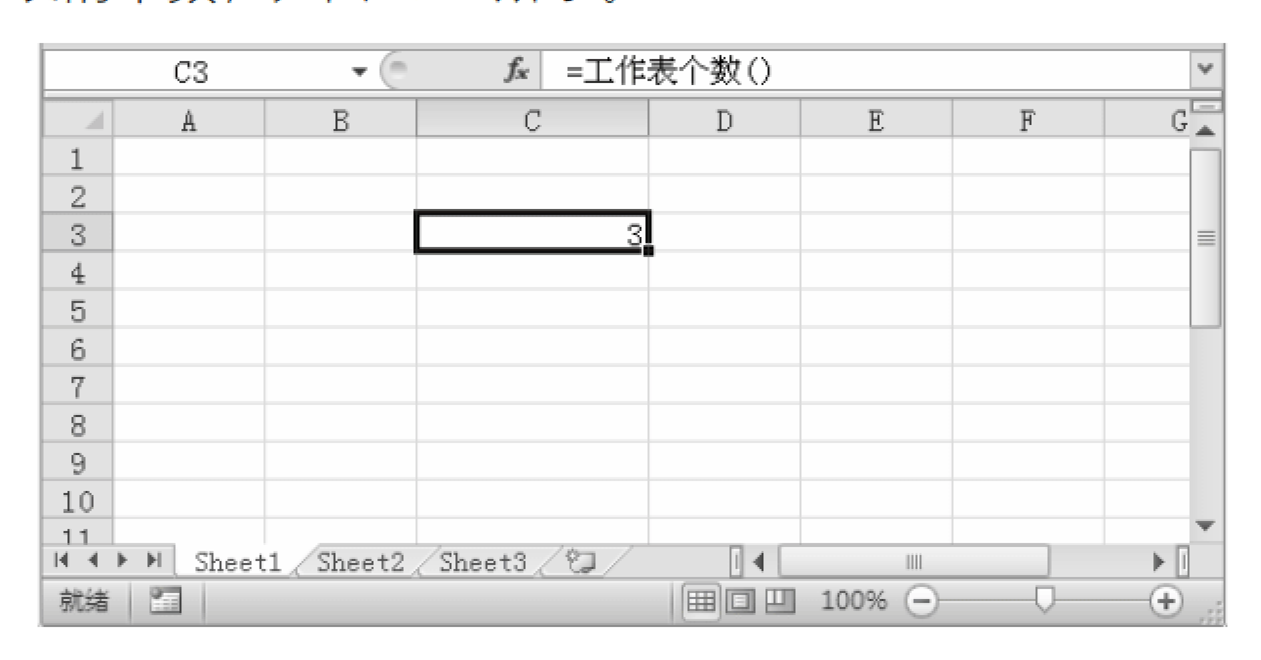


图 13-2　在工作表中使用自定义函数

13.1.2　VBA 中调用自定义函数

自定义函数不但可以在公式中使用，也可以被其他 VBA 程序调用，具体内容介绍如下。

【例 13.2】将上文中的自定义函数应用到一段新的程序代码中，实现工作簿中的工作表名称自动修改，修改为 1 月份、2 月份、3 月份……，具体操作步骤如下。

01 新建或打开一个 Excel 2010 工作簿，按 Alt+F11 组合键打开 VBA 代码窗口，在【工程资源管理器】中的任意位置右击，在弹出的快捷菜单中选择【插入】➤【模块】菜单命令，即可插入一个模块，双击该模块，在其中输入以下两段程序代码，如图 13-3 所示。

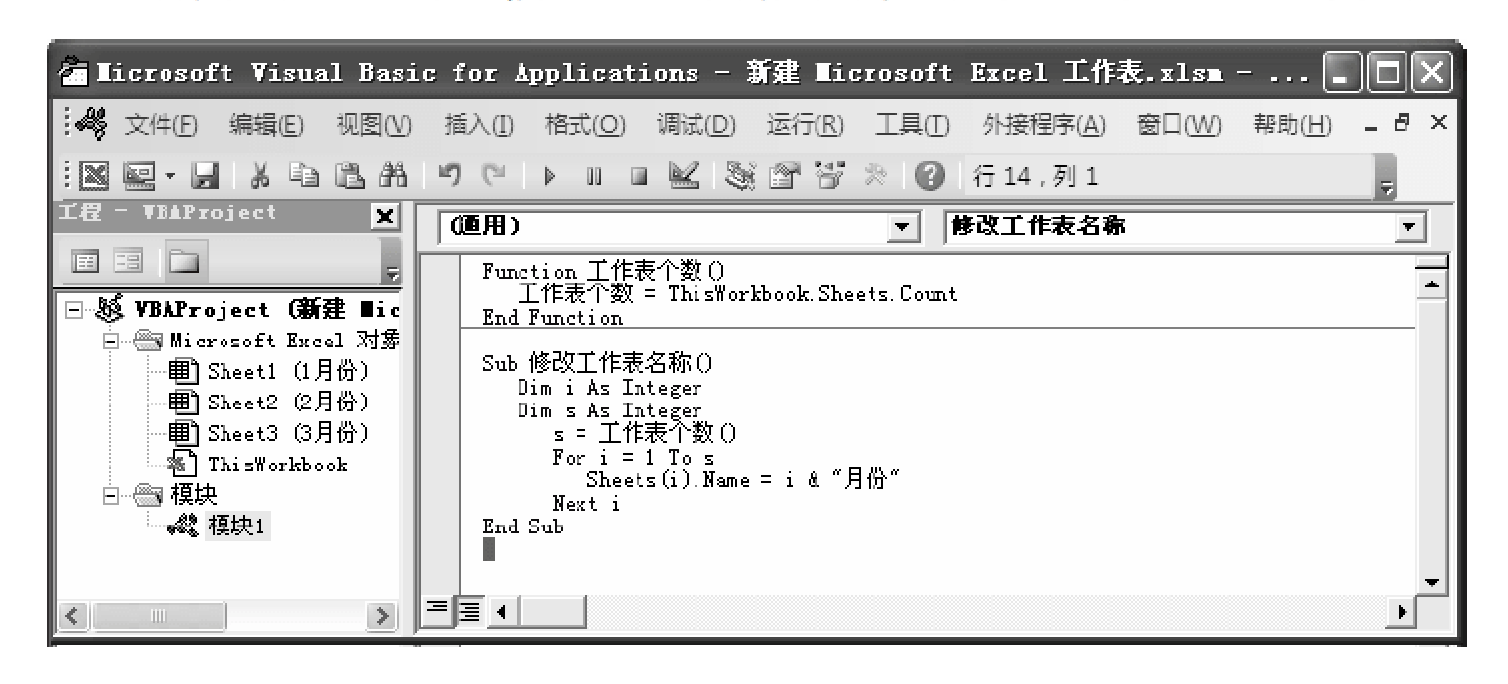


图 13-3　在其他程序中使用自定义函数

自定义函数代码如下：

```
Function 工作表个数()
   工作表个数 = ThisWorkbook.Sheets.Count
End Function
```

修改工作表名称代码如下：

```
Sub 修改工作表名称()
   Dim i As Integer
   Dim s As Integer
```

```
        s = 工作表个数()
        For i = 1 To s
           Sheets(i).Name =  i & "月份"
        Next i
  End Sub
```

02 单击【运行子过程/用户窗体】按钮▶，运行“修改工作表名称”程序，即可按要求将各工作表名称修改，如图 13-4 所示。

图 13-4　修改工作表名称

在代码中，“s=工作表个数（）”是利用自定义的“工作表个数”函数得到当前工作簿中工作表的个数，然后把结果赋予变量 s。

【注意】 在程序中使用函数时，如果没有参数，可以省略后面的“()”。



13.2 编写自定义函数

自定义函数根据需求不同，代码也有所差异，但是编写过程基本相似。下面以编写取得一个区域不重复值个数的函数为例，介绍自定义函数的编写过程。

13.2.1 自定义函数代码存放位置

内置函数在程序中有专门的存放位置，而自定义函数也需要为其指定一个存放位置。一般将其存放在添加的模块中，或者存放在指定对象（如工作表对象）的代码窗口中。

当然放置在不同位置的自定义函数是有差异的。在模块中的函数可以在任何一个程序中调用，而存放在对象代码窗口中的函数只能被本代码窗口内的程序所调用。如果要自定义一个在各个工作表公式中使用的函数，就必须在添加的新模块中编写。

本实例要编写一个取得指定区域中不重复值个数的函数，需要在新添加的模块中。

13.2.2 自定义函数代码的编制

新建一个 Excel 2010 工作簿，按 Alt+F11 组合键打开 VBA 代码窗口，在【工程资源管理

器】中任意位置右击，在弹出的快捷菜单中选择【插入】➢【模块】菜单命令，即可插入一个模块，双击该模块，进入该模块的代码窗口中，便可以编写自定义函数的代码。一般自定义函数的程序是以 Function 开始，以 End Function 结束，代码格式如下。

```
Function 函数名称 (自变量)
   程序代码
End Function
```

程序代码中各部分的含义介绍如下。

- 函数名称可以由字母、字符、数字和一些标点符号组成，但是第一个必须是字符。要注意的是，函数名称不能使用空格、句号、#、$、%、(、？、) 或!。
- 自变量的多少根据题目的需要而定。
- 程序代码中一定要包含函数名称的等式，如函数名称为“工作表个数”，那么程序中必须包含“工作表个数=可以返回值的代码或表达式”。

想要实现在指定的区域获取不重复值的个数，可在打开的模块代码窗口中输入以下代码，如图 13-5 所示。

```
Function 取不重复值个数(Rng As Range)
 Application.Volatile      ' 用于将用户自定义函数标记为易失性函数，无论何时在工作表的
'任意单元格中进行计算，易失性函数都必须重新进行计算。非易失性函数只在输入变量改变时才重新计算，
'若不用于计算工作表单元格的用户自定义函数中，则此方法无效
 Dim Mr As Range
 Dim X As Integer, K As Integer
 X = Application.CountA(Rng)
' 计算出给定区域非空单元格的字符或数字总个数，并赋予变量 x
  For Each Mr In Rng   '设置在给定的 Rng 单元格区域进行循环
    If Application.WorksheetFunction.CountIf(Rng, Mr) > 1 Then
' 调用工作表函数 CountIf 计算当前单元格在给定 Rng 单元格区域中的个数
     K = K + 1      '使用 K 对重复的个数进行累计
    End If
  Next Mr
取不重复值个数 = X - (K / 2)   ' 建立函数名称和计算内容关联，这句代码是必不可少的
End Function
```

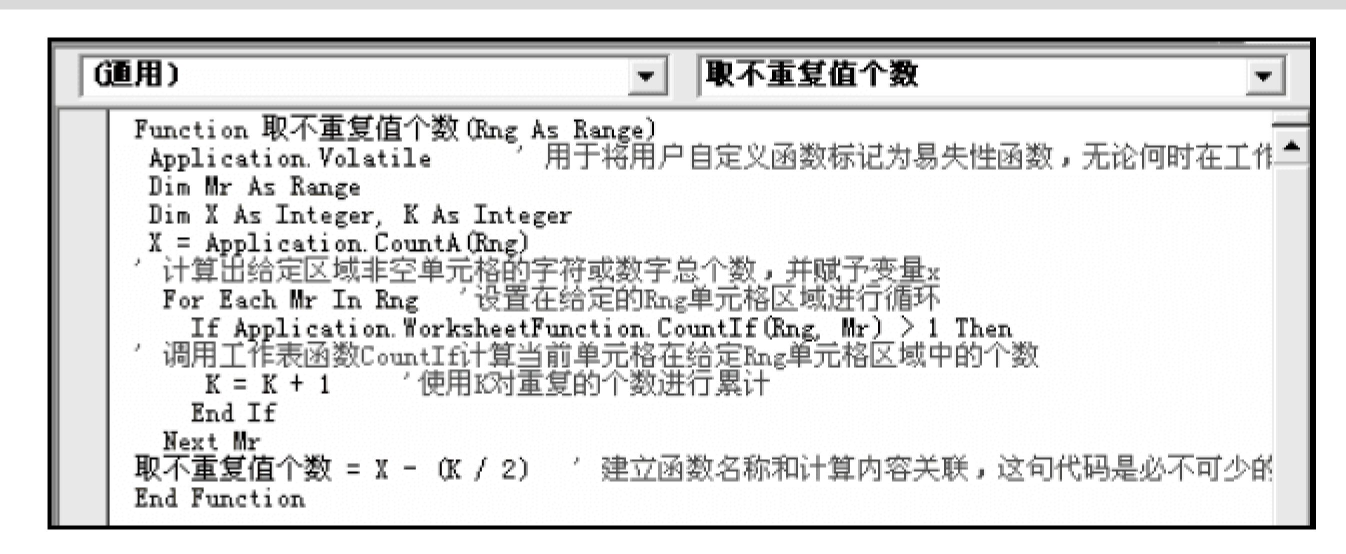


图 13-5　在模块中添加函数过程代码

代码中各部分含义解释如下。

- Function 取不重复值个数: Function 关键字说明这是函数过程。“取不重复值个数”是函数的名称。

- Rng As Range：设置自变量 Rng 并声明它的类型为单元格。
- X = Application.CountA（Rng）：这里调用了工作表函数 CountA，Application 后省略了 WorksheetFunction。
- X-（K/2）：是总个数减去重复的个数，再除以 2 的值，即所求的不重复值的个数。例如，A、B、C、D、A 总个数是 5（代码中 X 值），重复的值分别是第一个 A 和第二个 A，总个数是 2（代码中 K 的值），所以不重复值的个数为 5-2/2=4。

13.2.3 添加自定义函数的说明

在工作表中调用内置函数时，如果不清楚函数的用途，可以查看其说明文字。同时为了方便记忆自定义函数的功能，也可以为其添加一些说明文字，具体的操作步骤如下。

01 选择工作表功能区的【开发工具】选项卡，在【代码】选项组中单击【宏】按钮，打开【宏】对话框，如图 13-6 所示。

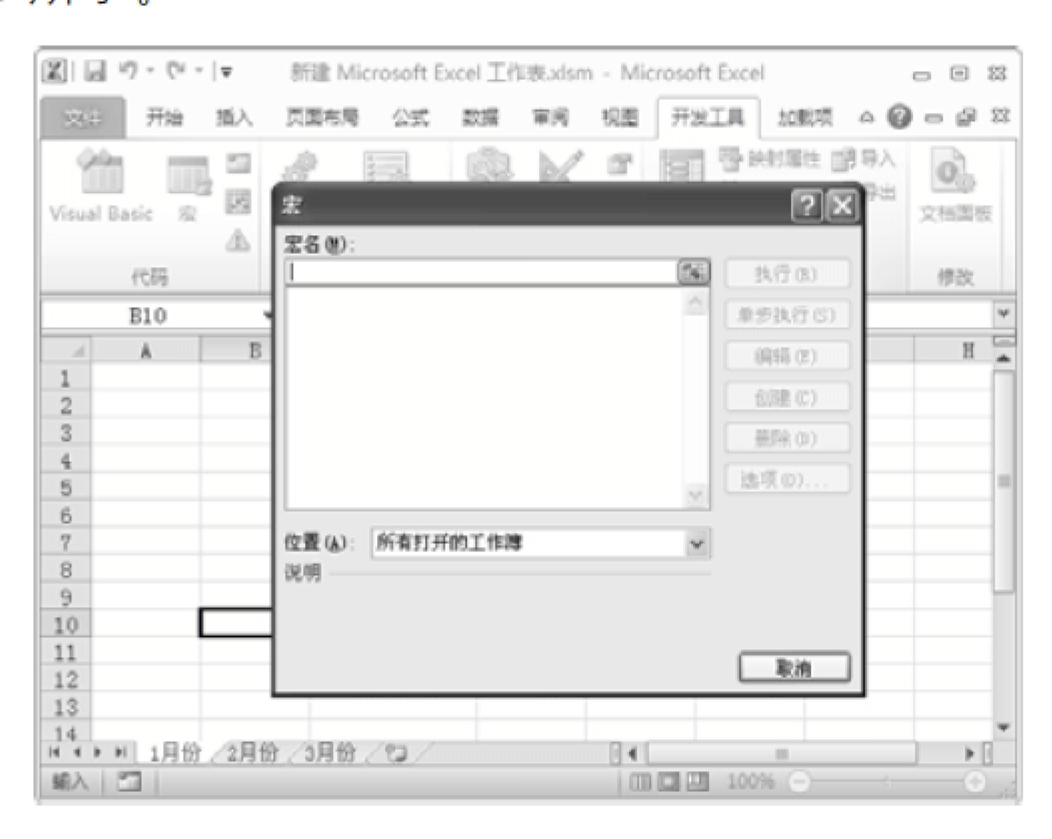


图 13-6 【宏】对话框

02 在【宏】对话框中的【宏名】文本框中输入自定义函数的名称"取不重复值个数"，如图 13-7 所示。

03 单击【选项】按钮，打开【宏选项】对话框，在【说明】文本框中输入函数的说明文字，如图 13-8 所示。

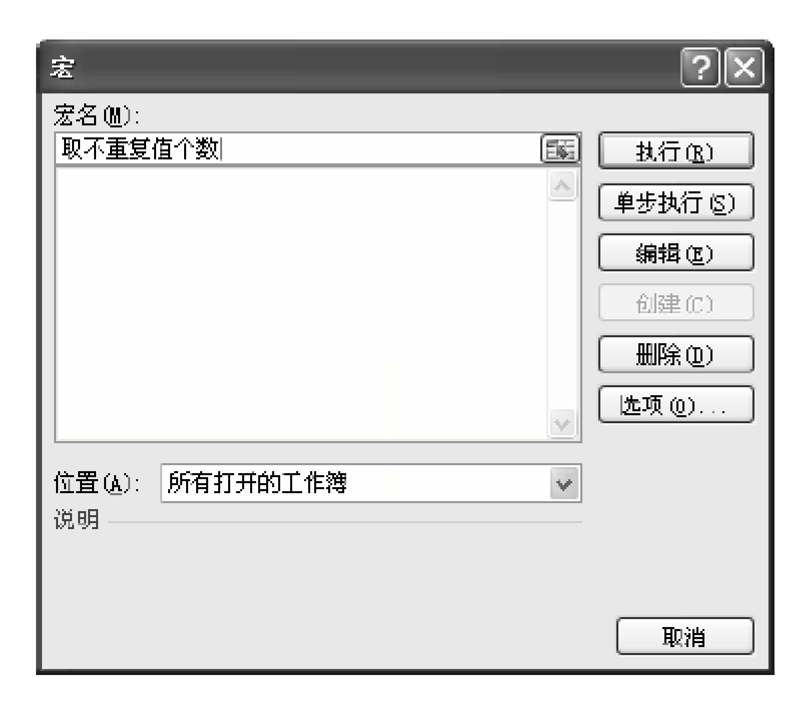


图 13-7 输入函数名称

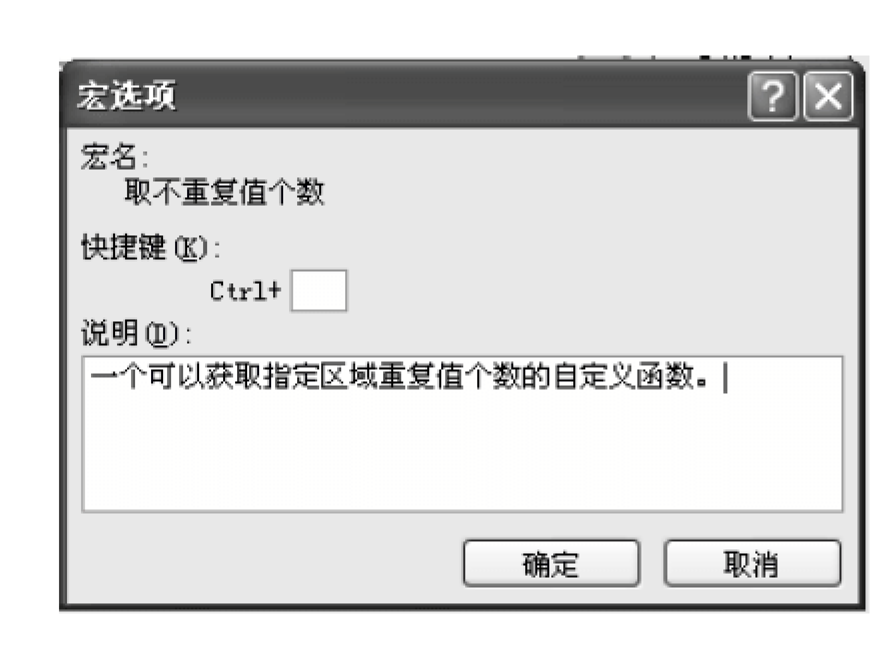


图 13-8 【宏选项】对话框

04 说明文字输入完成后，单击【确定】按钮，返回到【宏】对话框，即可看到添加的说明文字，如图 13-9 所示。

05 单击【取消】按钮，可完成说明文字的添加。

06 选择工作表功能区的【公式】选项卡，单击【函数库】选项组中的【插入函数】按钮，打开【插入函数】对话框，在【或选择类别】下拉列表框中选择【用户定义】选项，在【选择函数】列表框中即可显示已添加说明的自定义函数，同时在对话框的下方出现添加的函数说明，如图 13-10 所示。



图 13-9　显示添加的说明文字



图 13-10　添加后的函数说明

【提示】在插入工作表内置函数时，会提供该函数的编辑参数的说明，如图 13-11 所示。而自定义函数则无法添加这样的参数说明文字。



图 13-11　函数的参数说明

13.2.4 指定函数的类别

在 Excel 2010 中内置函数为了方便区分被分成了全部、财务、日期与时间等类别，而创建的自定义函数会自动分配到“用户定义”类别中。可以通过 VBA 代码将自定义函数划入

指定的函数类别中。首先在 VBA 代码窗口中插入一个模块，然后在该模块中输入如下代码程序，如图 13-12 所示。

```
Sub 指定函数类别()
Application.MacroOptions "取不重复值个数", Category:=4
End Sub
```

运行该程序后，就可以把自定义函数添加到“统计”函数类别中，如图 13-13 所示。

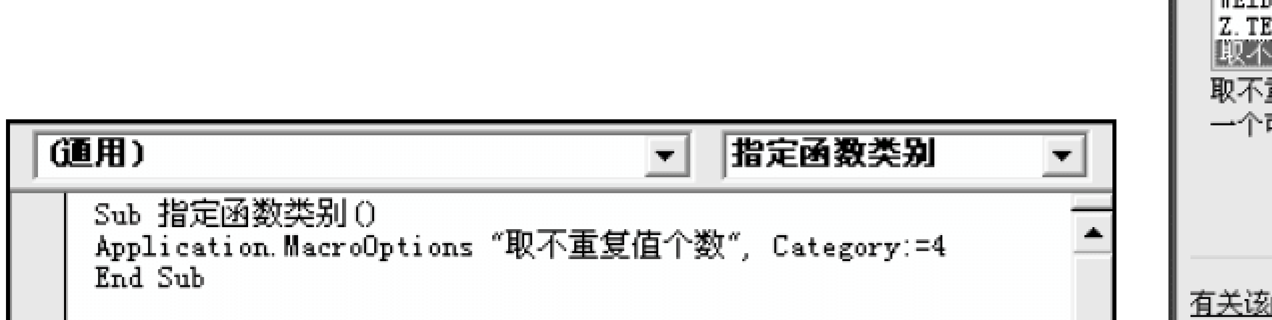


图 13-12　输入指定函数类别程序

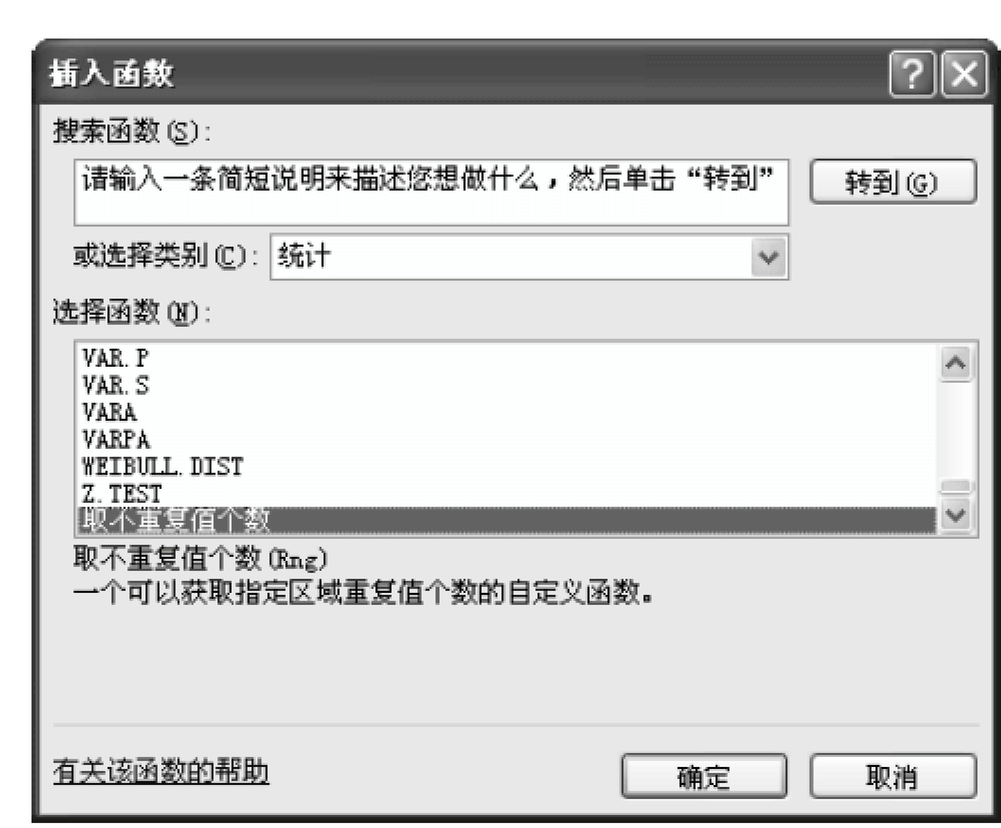


图 13-13　在统计函数中的自定义函数

代码中各部分含义解释如下：

- “取不重复值个数”：自定义函数名称。
- Category:=4：指定自定义函数的类别，4 代表“统计”函数。

表 13-1 列出了内置函数类别及编号。

表 13-1　内置函数类别及其编号

编号	函数类别	编号	函数类别
0	全部	5	查找和引用
1	财务	6	数据库
2	日期和时间	7	文本
3	数学和三角	8	逻辑
4	统计	9	信息



13.3 使用自定义函数

了解了自定义函数的基本知识和基本操作后，下面来介绍自定义函数的使用方法及在使用过程中所需要解决的问题。

13.3.1 在其他 Excel 2010 VBA 代码中使用

编写完成的自定义函数可以直接被当前工作簿内的其他程序调用，具体操作方法如下。

【例 13.3】用 Microsoft Excel 2010 提示框提示指定区域不重复值的个数，具体的操作步骤如下。

01 在上文创建的"模块 1"代码窗口中输入以下程序代码，如图 13-14 所示。

```
    Sub 提示不重复值的个数()
    Dim X As Integer
    X = 取不重复值的个数(Range("A1:A10"))  '在使用带参数的函数返回值时，应在该函数参数两边
'添加括号
    MsgBox "该区域不重复值的个数为：" & X & "个"
    End Sub
```

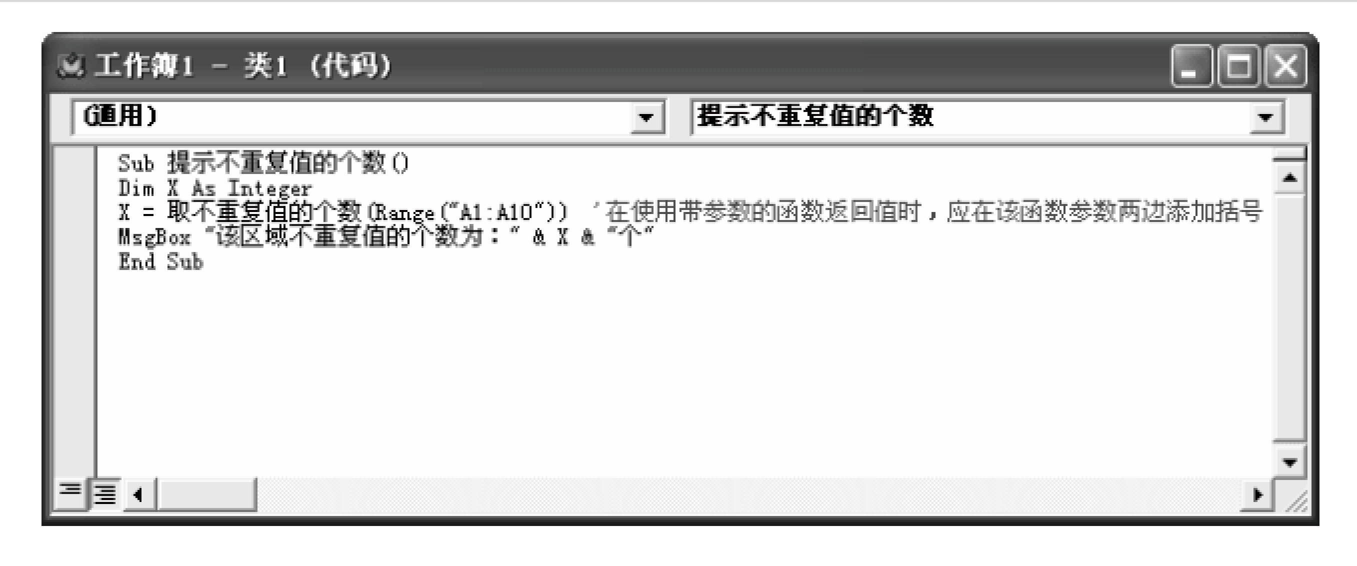


图 13-14　输入提示不重复个数的程序代码

02 运行该程序代码，即可弹出当前工作表的指定区域中不重复值的个数，如图 13-15 所示。

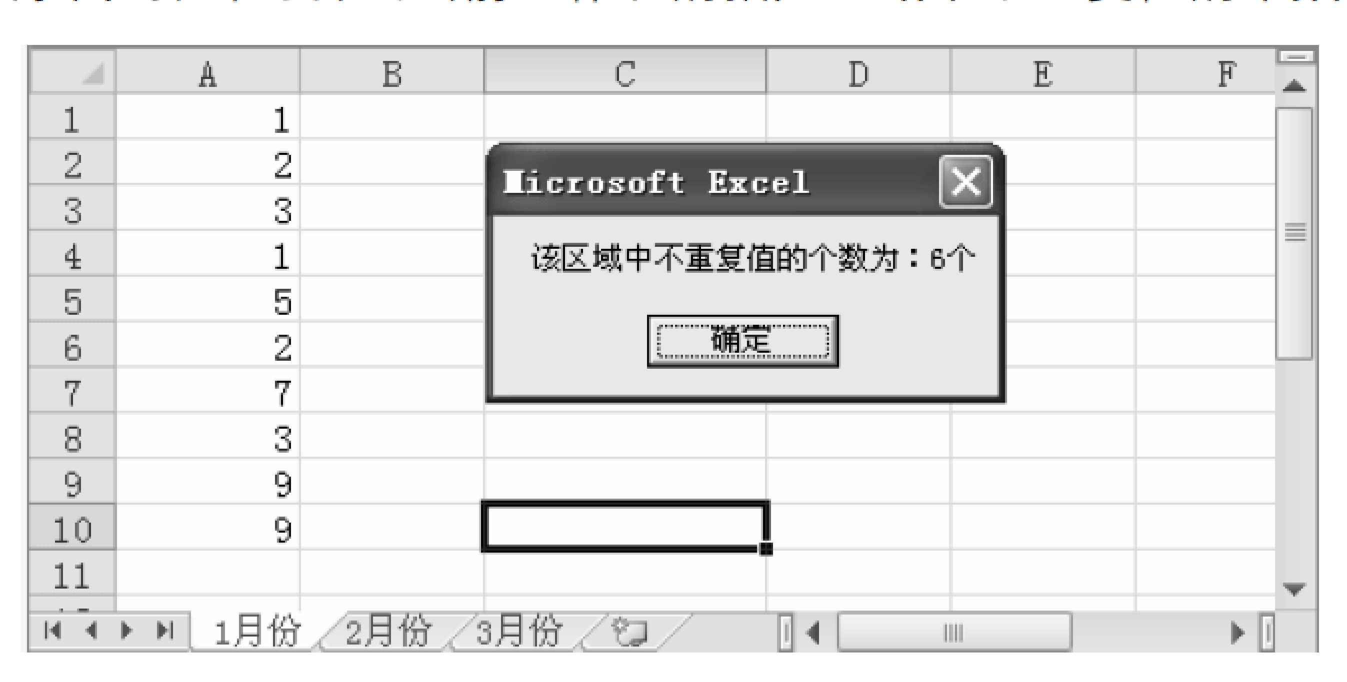


图 13-15　提示不重复值的个数

【提示】由以上内容看出，自定义的一个函数可以在多个任务中被调用，如果本实例中没有自定义函数，每一个操作任务的代码都将复杂很多。所以使用自定义函数可以大大简化代码的编辑。

13.3.2 在工作表公式中使用

在工作表中使用自定义函数，与内置函数的使用方法相同，具体操作方法如下。

【例 13.4】在 C2 单元格中计算 A1：A10 单元格区域中不重复值的个数，具体的操作步骤如下。

01 在 C2 单元格中输入公式〝=取不重复值个数（A1:A10）〞，如图 13-16 所示。

SUM　=取不重复值个数(A1:A10)

	A	B	C	D	E
1	1		A1:A10中不重复值的个数		
2	2		=取不重复值个数(A1:A10)		
3	3				
4	1				
5	5				
6	2				
7	7				
8	3				
9	9				
10	9				
11					
12					
13					

1月份　2月份　3月份

图 13-16　输入计算公式

02 按下 Enter 键，即可计算出 A1:A10 单元格区域中不重复值的个数，如图 13-17 所示。

C2　=取不重复值个数(A1:A10)

	A	B	C	D	E
1	1		A1:A10中不重复值的个数		
2	2		6		
3	3				
4	1				
5	5				
6	2				
7	7				
8	3				
9	9				
10	9				
11					
12					
13					

1月份　2月份　3月份

图 13-17　在工作表中使用自定义函数

【提示】也可以通过单击【公式】选项卡中的【函数库】选项组中的【插入函数】按钮来输入自定义的函数。

13.3.3 自定义函数的公用

一个自定义的函数并不一定只能在当前工作簿使用，也可以应用到其他打开的工作簿中。这需要创建加载宏文件，并在工作簿中加载该加载宏。

【例 13.6】使用加载宏的方法，让自定义函数“取不重复值个数”实现公用。具体操作步骤如下。

01 选择工作簿功能区的【文件】选项卡，然后选择【信息】选项组，单击右侧列表框的【属性】下三角按钮，在弹出的下拉列表中选择【高级属性】选项，如图 13-18 所示。

02 打开该工作簿的属性对话框，选择【摘要】选项卡，在【标题】文本框中输入〝取不重

复值个数加载宏”，在【备注】文本框中输入关于该加载宏的详细说明，输入完毕后单击【确定】按钮，即可保存该加载宏提示信息，如图 13-19 所示。

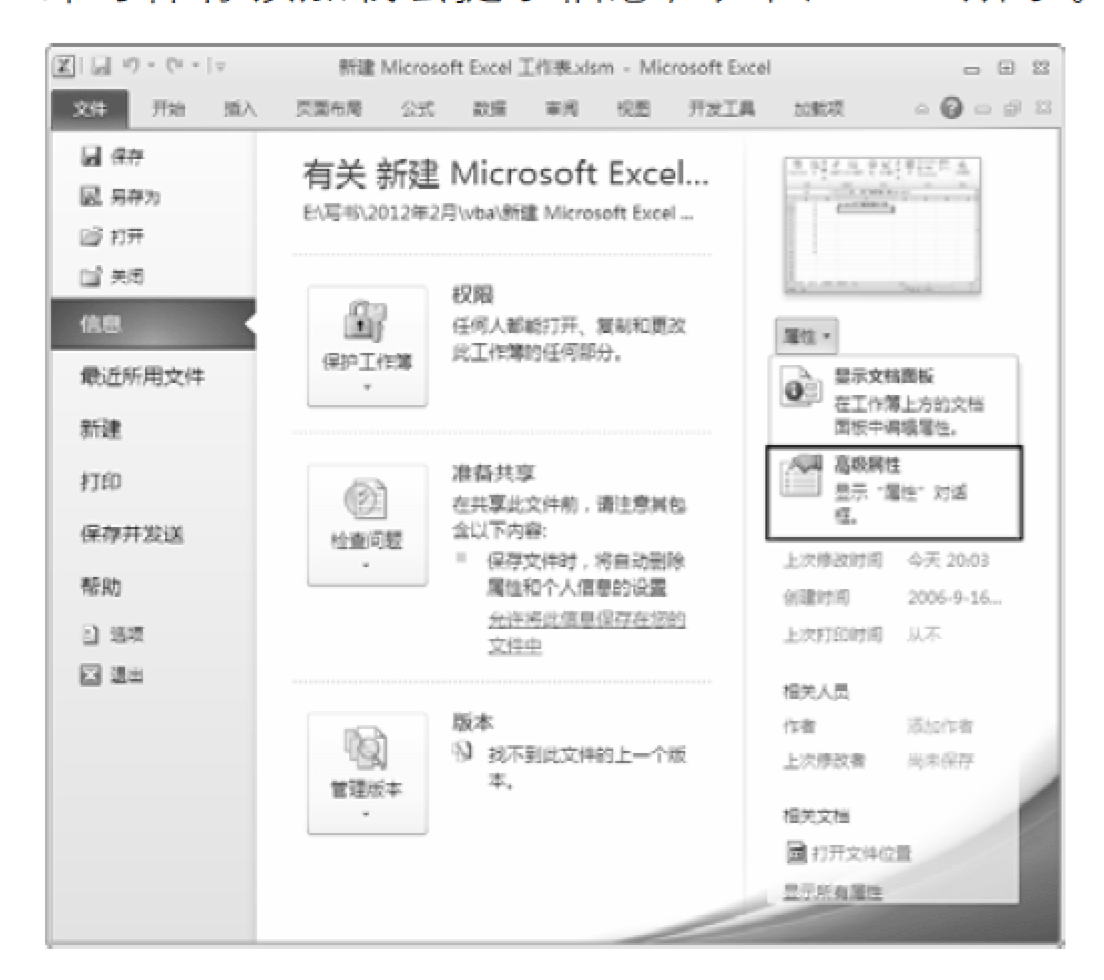


图 13-18　【信息】选项组



图 13-19　工作簿的属性对话框

03 打开含有自定义函数代码的文件并选择功能区【文件】➤【另存为】菜单命令，打开【另存为】对话框，从中输入加载宏的名称（名称要利于辨认该自定义函数），并在【保存类型】下拉列表框中选择保存类型为“Excel 2010 加载宏”选项，单击【保存】按钮即可保存成功，如图 13-20 所示。

04 选择 Excel 2010 功能区的【开发工具】选项卡，单击【加载项】按钮，打开【加载宏】对话框，如图 13-21 所示。



图 13-20　【另存为】对话框

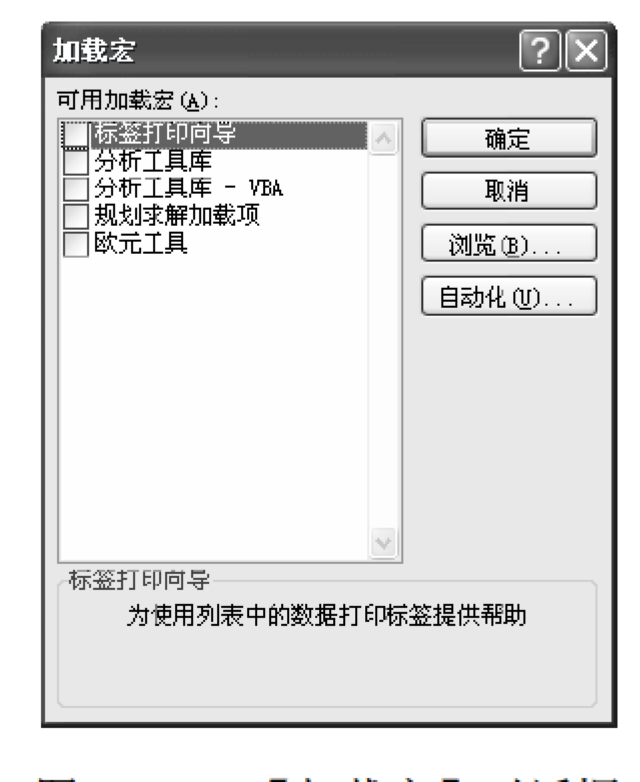


图 13-21　【加载宏】对话框

【提示】Windows XP 系统下加载宏文件的默认保存位置为 C:\Documents and Settings\Administrator\Application Data\Microsoft\AddIns 文件夹下。

05 在【加载宏】对话框中单击【浏览】按钮，即可打开【浏览】对话框，从中选择已保存的加载宏文件，单击【确定】按钮，如图 13-22 所示。



06 返回【加载宏】对话框，在【可用加载宏】列表中便会出现【取不重复值个数加载宏】选项，并且在对话框底部还有步骤 2 中添加的说明，如图 13-23 所示，单击【确定】按钮即可完成自定义函数的加载过程。



图 13-22 【浏览】对话框

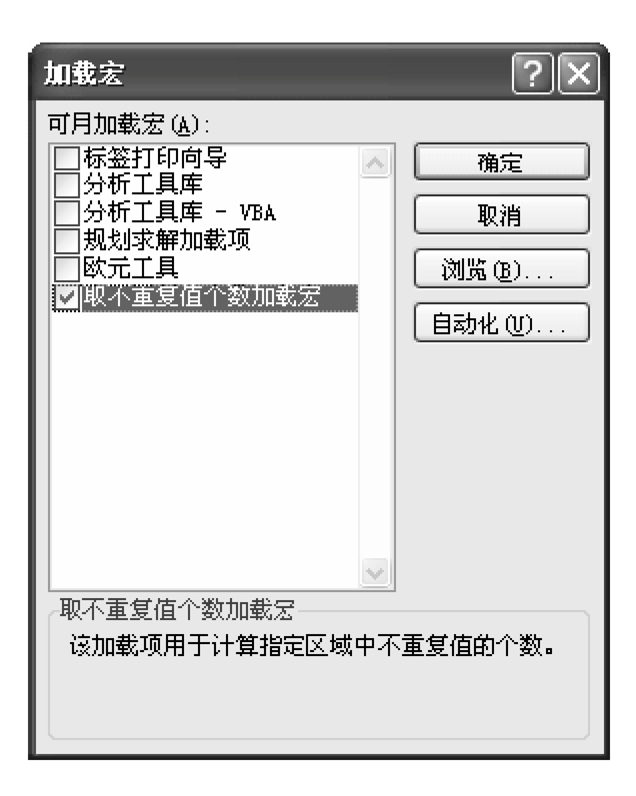


图 13-23 显示加载的宏名称



13.4 综合实战 1——定义计算员工奖金的函数

奖金计算在工资统计中应用非常广泛，特别是销售行业，为了促进销售人员的工作积极性，销售部门往往会制定一系列的销售业绩奖金制度。下面来分析一个典型的奖金计算案例。

销售经理制定了销售业绩奖金制度，奖金发放的标准奖金率如表 13-2 所示。

表 13-2 标准奖金率

月销售额	奖金率	月销售额	奖金率
≤3000	4%	15001~30000	12%
3000~8000	6%	30001~50000	16%
8001~15000	9%	>50000	20%

同时，为了鼓励员工持续地为公司工作，工龄越长对奖金越有利，具体规定为：参与计算的奖金率等于标准奖金率加上工龄一半的百分数。例如，一个工龄为 6 年的员工，标准奖金率为 6%时，参与计算的奖金率则为 6%+（6/2）%＝9%。

首先在 Excel 2010 中制作好如图 13-24 所示的 Sheet1 工作表。然后计算方法分析如下。

	A	B	C	D	E	F
1	姓名	销售业绩	工龄	奖金		
2	陈柯	¥17,890.00	4			
3	李丽	¥9,600.00	1			
4	孔玲	¥9,580.00	6			
5	张云	¥6,520.00	2			
6	刘增	¥3,998.00	2			
7	张军	¥3,750.00	3			
8	薛凯	¥3,560.00	6			
9	崔艳	¥3,200.00	5			
10	张红	¥3,000.00	5			
11	李艳	¥2,650.00	2			
12	张磊	¥2,580.00	1			
13	郑光	¥2,250.00	5			
14	韩少华	¥1,950.00	3			
15	李立威	¥1,800.00	3			
16	王杰	¥1,760.00	2			

Sheet1　Sheet2　Sheet3

图 13-24　制作计算奖金的员工工作表

若不考虑工龄对奖金率的影响，那么可以利用嵌套使用 IF 函数，在 D2 单元格输入公式“=IF（B2<=3000,B2*4%,IF（B2<=8000,B2*6%,IF（B2<=15000,B2*9%,IF（B2<=30000,B2*12%,IF（B2<=50000,B2*16%,B2*20%)))))”进行计算。

但是，可以看出以上公式有两个明显的弊端。

（1）公式看起来太烦琐，不容易理解，而且 IF 函数最多只能嵌套 7 层，万一奖金率超过 7 个，那么这个方法就没用了。

（2）由于没有考虑工龄，所以该方法不能算是解决问题了，如果把工龄考虑到上述公式中，这样公式就会显得更加冗长烦琐，以后的管理与调整都很不方便。

所以还是使用自定义函数进行计算比较简捷。这里与上面的实例有所不同的是，该自定义函数使用了两个参数，具体的操作步骤如下。

01 选择 Excel 2010 工作表的功能区【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮或按 Alt+F11 组合键，打开 Excel 2010 VBA 代码窗口。在该窗口中的【工程资源管理器】窗口中任意位置右击，在弹出的快捷菜单中选择【插入】➤【模块】菜单命令，插入一个模块，双击该模块，并在该模块中输入如下代码程序，如图 13-25 所示。

```
Function REWARD(sales, years) As Double
Const r1 As Double = 0.04
Const r2 As Double = 0.06
Const r3 As Double = 0.9
Const r4 As Double = 0.12
Const r5 As Double = 0.16
Const r6 As Double = 0.19
    Select Case sales
      Case Is <= 3000
         REWARD = sales * (r1 + years / 200)
      Case Is <= 8000
```

```
            REWARD = sales * (r2 + years / 200)
        Case Is <= 15000
            REWARD = sales * (r3 + years / 200)
        Case Is <= 30000
            REWARD = sales * (r4 + years / 200)
        Case Is <= 50000
            REWARD = sales * (r5 + years / 200)
        Case Is > 50000
            REWARD = sales * (r6 + years / 200)
    End Select
End Function
```

```
(通用)                                TAX
Function REWARD(sales, years) As Double
Const r1 As Double = 0.04
Const r2 As Double = 0.07
Const r3 As Double = 0.1
Const r4 As Double = 0.13
Const r5 As Double = 0.16
Const r6 As Double = 0.19
    Select Case sales
        Case Is <= 2800
            REWARD = sales * (r1 + years / 200)
        Case Is <= 7900
            REWARD = sales * (r2 + years / 200)
        Case Is <= 15000
            REWARD = sales * (r3 + years / 200)
        Case Is <= 30000
            REWARD = sales * (r4 + years / 200)
        Case Is <= 50000
            REWARD = sales * (r5 + years / 200)
        Case Is > 50000
            REWARD = sales * (r6 + years / 200)
    End Select
End Function
```

图 13-25　输入代码

【提示】在上述代码中，自定义了一个名为 REWARD 的函数，它包含两个参数：销售额 sales 和工龄 years。常量 r1~r6 分别存放着各个等级的奖金率，这样处理的好处是当奖金率调整时，修改非常方便。同时，函数的层次结构比前面的公式清晰，使用户容易理解函数的功能。此外，当奖金率超过 7 个时，用自定义函数的方法仍然可以轻松处理。

02 函数自定义完成后，返回 Excel 2010 工作表窗口，在 D2 单元格中输入"=reward(B2,C2)"，如图 13-26 所示。

	A	B	C	D	E	F
1	姓名	销售业绩	工龄	奖金		
2	陈柯	¥17,890.00	4	=reward(B2,C2)		
3	李丽	¥9,600.00	1			
4	孔玲	¥9,580.00	6			
5	张云	¥6,520.00	2			
6	刘增	¥3,998.00	2			
7	张军	¥3,750.00	3			
8	薛凯	¥3,560.00	6			
9	崔艳	¥3,200.00	5			
10	张红	¥3,000.00	5			
11	李艳	¥2,650.00	2			
12	张磊	¥2,580.00	1			
13	郑光	¥2,250.00	5			
14	韩少华	¥1,950.00	3			
15	李立威	¥1,800.00	3			
16	王杰	¥1,760.00	2			

Sheet1 / Sheet2 / Sheet3

图 13-26　在 D2 单元格中输入公式

03 按 Enter 键即可计算出第一个员工的奖金，如图 13-27 所示。

	A	B	C	D	E
1	姓名	销售业绩	工龄	奖金	
2	陈柯	¥17,890.00	4	¥2,504.60	
3	李丽	¥9,600.00	1		
4	孔玲	¥9,580.00	6		
5	张云	¥6,520.00	2		
6	刘增	¥3,998.00	2		
7	张军	¥3,750.00	3		
8	薛凯	¥3,560.00	6		

图 13-27　计算出第一个员工的奖金

04 然后利用公式填充柄复制该公式到后面的单元格，即可完成对其他员工奖金的计算。



13.5　综合实战 2——定义员工个人调节税的函数

计算个人调节税使用也比较广泛，为其设置自定义函数的操作内容如下。

首先，假设个人调节税的收缴标准是：工资小于等于 800 元的免征调节税，工资 800 元以上至 1500 元的超过部分按 5%的税率征收，1500 元以上至 2000 元的超过部分按 8%的税率征收，高于 2000 元的超过部分按 20%的税率征收。

例如，在 Sheet1 工作表的 A、B、C、D 列中分别存放“姓名”、“总工资”、“调节税”、“税后工资”字段数据，如图 13-28 所示。

	A	B	C	D	E	F
1	姓名	总工资	调节税	税后工资		
2	刘增	¥750.00				
3	张云	¥1,250.00				
4	张军	¥1,750.00				
5	郑光	¥2,250.00				
6	张红	¥3,000.00				
7	孔玲	¥1,300.00				
8	王杰	¥1,760.00				
9	张磊	¥2,580.00				
10	崔艳	¥1,479.00				
11	薛凯	¥3,560.00				
12	李立威	¥890.00				
13	李丽	¥950.00				
14	韩少华	¥1,950.00				
15	李艳	¥2,650.00				
16	陈柯	¥880.00				
17						

Sheet1 Sheet2 Sheet3

图 13-28　Sheet1 工作表



平时使用较多的方法是借助嵌套使用 IF 函数计算，例如在 C2 单元格输入公式“=IF（B2<=800,0,IF（B2<=1500,（B2-800）*0.05,IF（B2<=2000,700*0.05+（B2-1500）*0.08,700*0.05+500*0.08+（B2-2000）*0.2)”，然后通过填充柄复制公式到 C 列的其余单元格，即可自动计算出调节税，如图 13-29 所示。

C2 =IF(B2<=800, 0, IF(B2<=1500, (B2-800)*

	A	B	C	D	E	F
1	姓名	总工资	调节税	税后工资		
2	刘增	¥750.00	¥0.00			
3	张云	¥1,250.00	¥22.50			
4	张军	¥1,750.00	¥55.00			
5	郑光	¥2,250.00	¥125.00			
6	张红	¥3,000.00	¥275.00			
7	孔玲	¥1,300.00	¥25.00			
8	王杰	¥1,760.00	¥55.80			
9	张磊	¥2,580.00	¥191.00			
10	崔艳	¥1,479.00	¥33.95			
11	薛凯	¥3,560.00	¥387.00			
12	李立威	¥890.00	¥4.50			
13	李丽	¥950.00	¥7.50			
14	韩少华	¥1,950.00	¥71.00			
15	李艳	¥2,650.00	¥205.00			
16	陈柯	¥880.00	¥4.00			

图 13-29　填充公式

可以看到，公式可以解决个人调节税统计的问题，但是为什么还要使用自定义函数的方法呢？很显然，使用公式看起来太烦琐，不便于理解和管理；而且公式的处理能力在面对稍微复杂一些的问题时便失去效用。假设调节税的税率标准会根据年龄的不同而改变，那么公式可能就没用了。

下面就通过此例介绍使用自定义函数的全过程，具体的操作步骤如下。

01 选择 Excel 2010 工作表功能区的【开发工具】选项卡，单击【代码】选项组中的【Visual Basic】按钮或按 Alt+F11 组合键，打开 Excel 2010 VBA 代码窗口。在【工程资源管理器】窗口中任意位置右击，在弹出的快捷菜单中选择【插入】➢【模块】菜单命令，插入一个模块，双击该模块，并在该模块中输入如下代码程序，如图 13-30 所示。

```
Function TAX(salary)
    Const r1 As Double = 0.05
    Const r2 As Double = 0.08
    Const r3 As Double = 0.2
    Select Case salary
    Case Is <= 800
    TAX = 0
    Case Is <= 1500
    TAX = (salary - 800) * r1
    Case Is <= 2000
    TAX = (1500 - 800) * r1 + (salary - 1500) * r2
    Case Is > 2000
```

```
        TAX = (1500 - 800) * r1 + (2000 - 1500) * r2 + (salary - 2000) * r3
    End Select
End Function
```

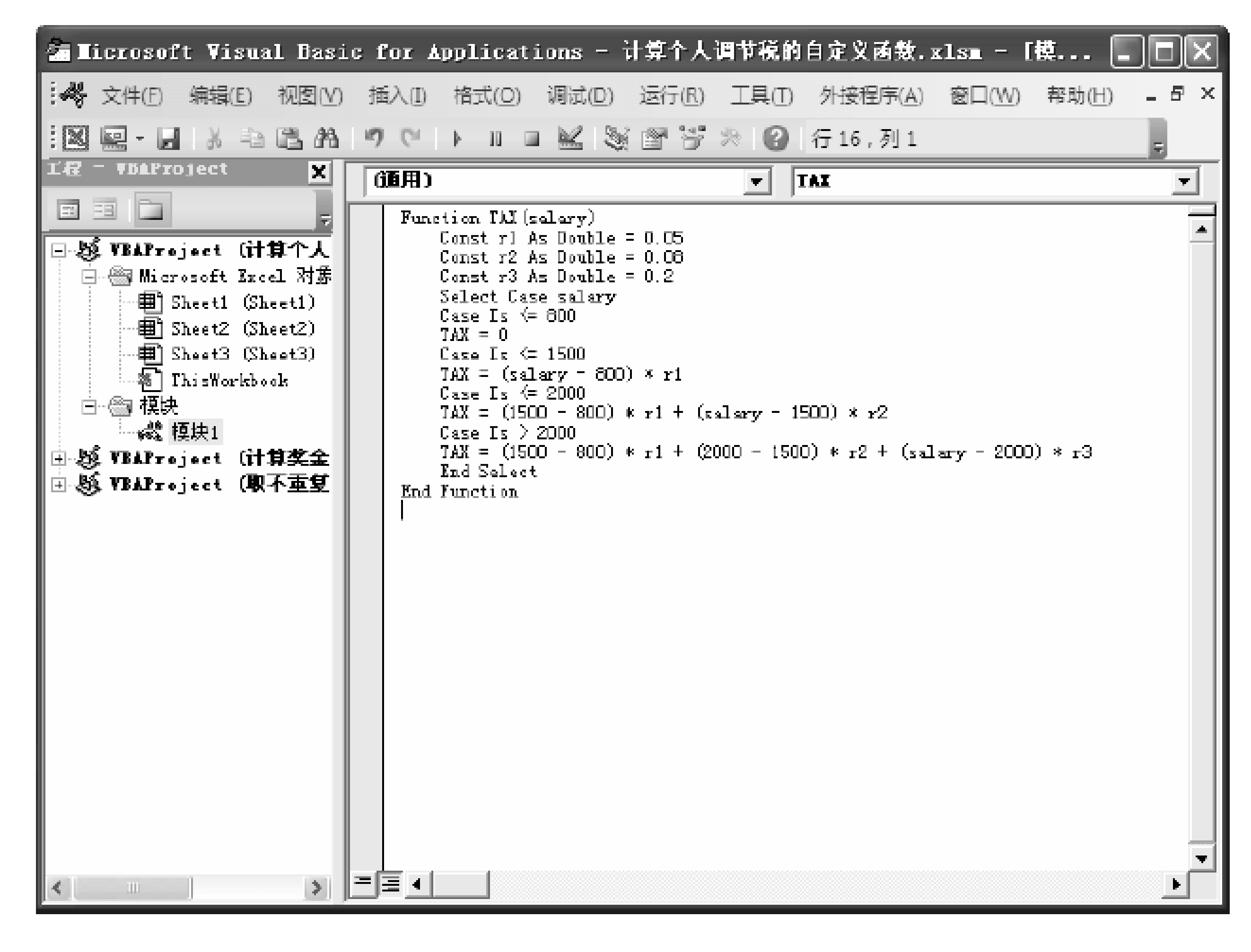


图 13-30　输入自定义函数代码

02 函数自定义完成后，返回 Excel 2010 工作表窗口，在 C2 单元格中输入公式"=TAX(B2)"，如图 13-31 所示。

SUM　=TAX(B2)

	A	B	C	D	E	F
1	姓名	总工资	调节税	税后工资		
2	刘增	¥750.00	=TAX(B2)			
3	张云	¥1, 250.00				
4	张军	¥1, 750.00				
5	郑光	¥2, 250.00				
6	张红	¥3, 000.00				
7	孔玲	¥1, 300.00				
8	王杰	¥1, 760.00				
9	张磊	¥2, 580.00				
10	崔艳	¥1, 479.00				
11	薛凯	¥3, 560.00				
12	李立威	¥890.00				
13	李丽	¥950.00				
14	韩少华	¥1, 950.00				
15	李艳	¥2, 650.00				
16	陈柯	¥880.00				

图 13-31　输入公式

03 按 Enter 键，计算出了第一个员工应付的个人调节税，如图 13-32 所示。

C2 =TAX(B2)

姓名	总工资	调节税	税后工资
刘增	¥750.00	¥0.00	
张云	¥1,250.00		
张军	¥1,750.00		
郑光	¥2,250.00		
张红	¥3,000.00		
孔玲	¥1,300.00		
王杰	¥1,760.00		
张磊	¥2,580.00		
崔艳	¥1,479.00		
薛凯	¥3,560.00		
李立威	¥890.00		

Sheet1 Sheet2 Sheet3

图 13-32　计算出第一个员工应付的个人调节税

04 用公式填充柄复制公式到其他单元格，完成所有员工个人调节税的计算，如图 13-33 所示。

C2 =TAX(B2)

姓名	总工资	调节税	税后工资
刘增	¥750.00	¥0.00	
张云	¥1,250.00	¥22.50	
张军	¥1,750.00	¥55.00	
郑光	¥2,250.00	¥125.00	
张红	¥3,000.00	¥275.00	
孔玲	¥1,300.00	¥25.00	
王杰	¥1,760.00	¥55.80	
张磊	¥2,580.00	¥191.00	
崔艳	¥1,479.00	¥33.95	
薛凯	¥3,560.00	¥387.00	
李立威	¥890.00	¥4.50	
李丽	¥950.00	¥7.50	
韩少华	¥1,950.00	¥71.00	
李艳	¥2,650.00	¥205.00	

Sheet1 Sheet2 Sheet3

图 13-33 计算所有员工应付的个人调节税

从自定义函数的代码中可以看出，用这种方式，自定义函数的功能易于理解，同时如果税率改变，相应地变化 r1、r2、r3 的值即可。



13.6 综合实战 3——按颜色求和与计数

在 Excel 2010 中，如果要以单元格背景色为条件来进行数据的求和与计数是件困难的事，因为 Excel 2010 并没有提供相关的功能。而使用自定义函数却可以完成该功能，具体操作步骤如下。

01 在 Sheet1 工作表中，按如图 13-34 所示的效果将 A1:A10 单元格区域中分别填充背景颜色，并在其他单元格中输入相应的数据。

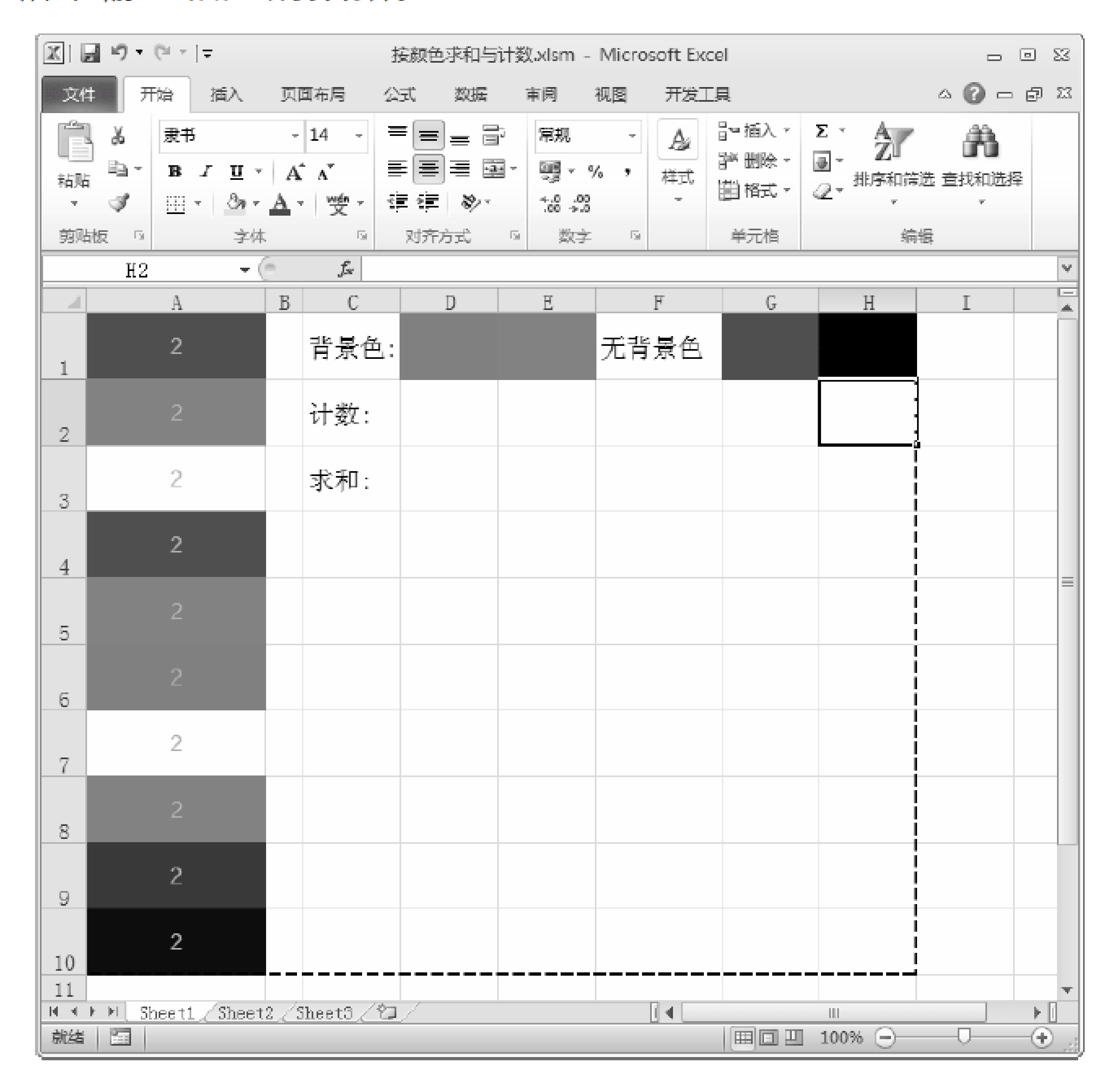


图 13-34　填充背景颜色

02 选择 Excel 2010 工作表功能区的【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮或按 Alt+F11 组合键，即可打开 Excel 2010 VBA 代码窗口。在【工程资源管理器】窗口中任意位置右击，在弹出的快捷菜单中选择【插入】➢【模块】菜单命令，插入一个模块，双击该模块，并在该模块中输入如下两段程序代码，如图 13-35 所示。

计数自定义函数代码：

```
Function CountColor(col As Range, countrange As Range) As Integer
    Dim icell As Range
    Application.Volatile
    For Each icell In countrange
        If icell.Interior.ColorIndex = col.Interior.ColorIndex Then
            CountColor = CountColor + 1
        End If
    Next icell
End Function
```

求和自定义函数代码：

```
Function SumColor(col As Range, sumrange As Range) As Integer
    Dim icell As Range
    Application.Volatile
    For Each icell In sumrange
        If icell.Interior.ColorIndex = col.Interior.ColorIndex Then
```




```
            SumColor = Application.Sum(icell) + SumColor
        End If
    Next icell
End Function
```

```
(通用)                                   SumColor

Function CountColor(col As Range, countrange As Range) As Integer
    Dim icell As Range
    Application.Volatile
    For Each icell In countrange
        If icell.Interior.ColorIndex = col.Interior.ColorIndex Then
            CountColor = CountColor + 1
        End If
    Next icell
End Function

Function SumColor(col As Range, sumrange As Range) As Integer
    Dim icell As Range
    Application.Volatile
    For Each icell In sumrange
        If icell.Interior.ColorIndex = col.Interior.ColorIndex Then
            SumColor = Application.Sum(icell) + SumColor
        End If
    Next icell
End Function
```

图 13-35　输入自定义函数代码

03 自定义函数输入完毕后，返回工作表窗口中。在 D2 单元格中输入公式“=countcolor(D$1,$A$1:$A$10)”，如图 13-36 所示。

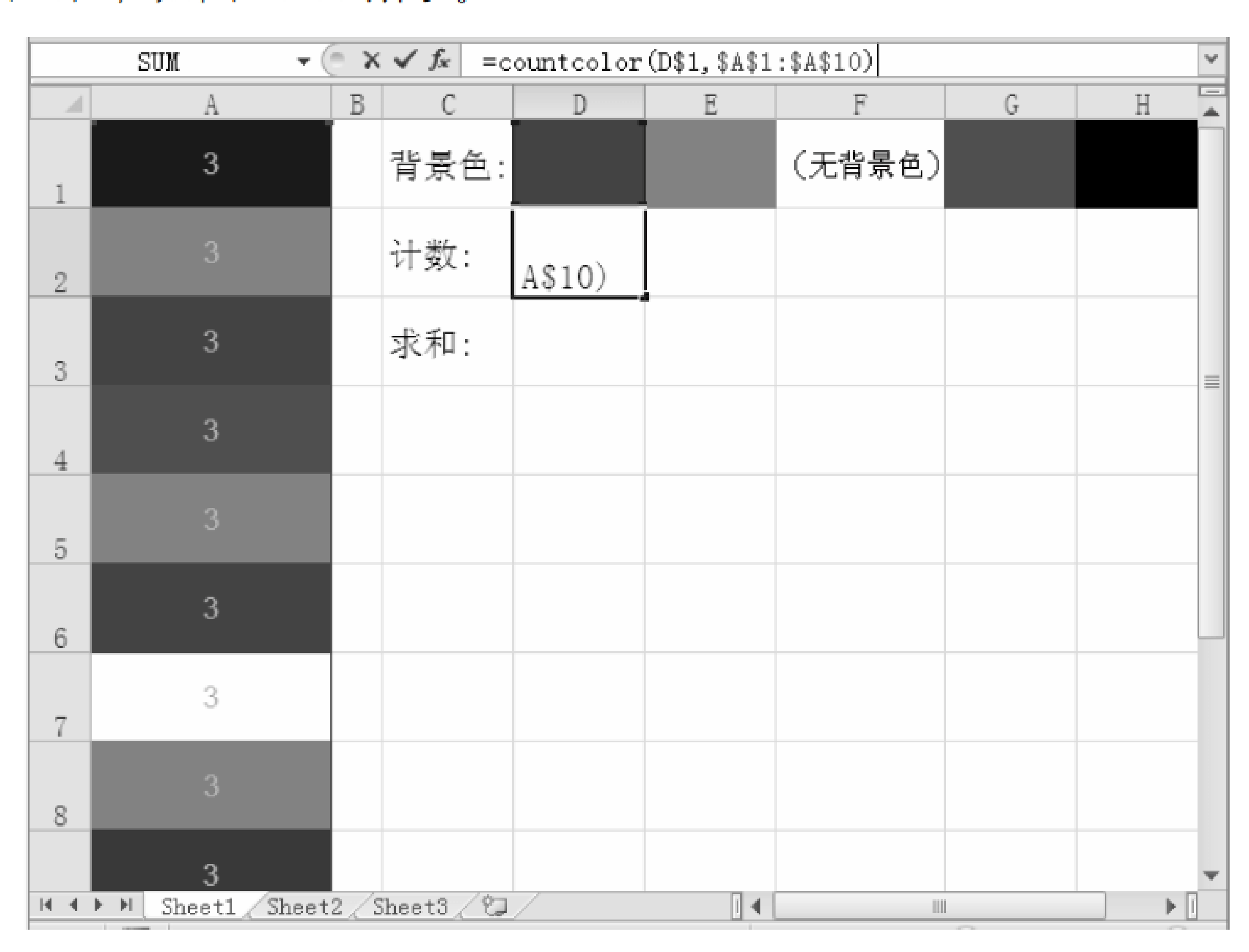


图 13-36　输入计数公式

04 按 Enter 键，即可统计出 A1:A10 单元格区域中背景颜色与 D1 单元格中的背景颜色相同（灰色）的单元格个数，如图 13-37 所示。

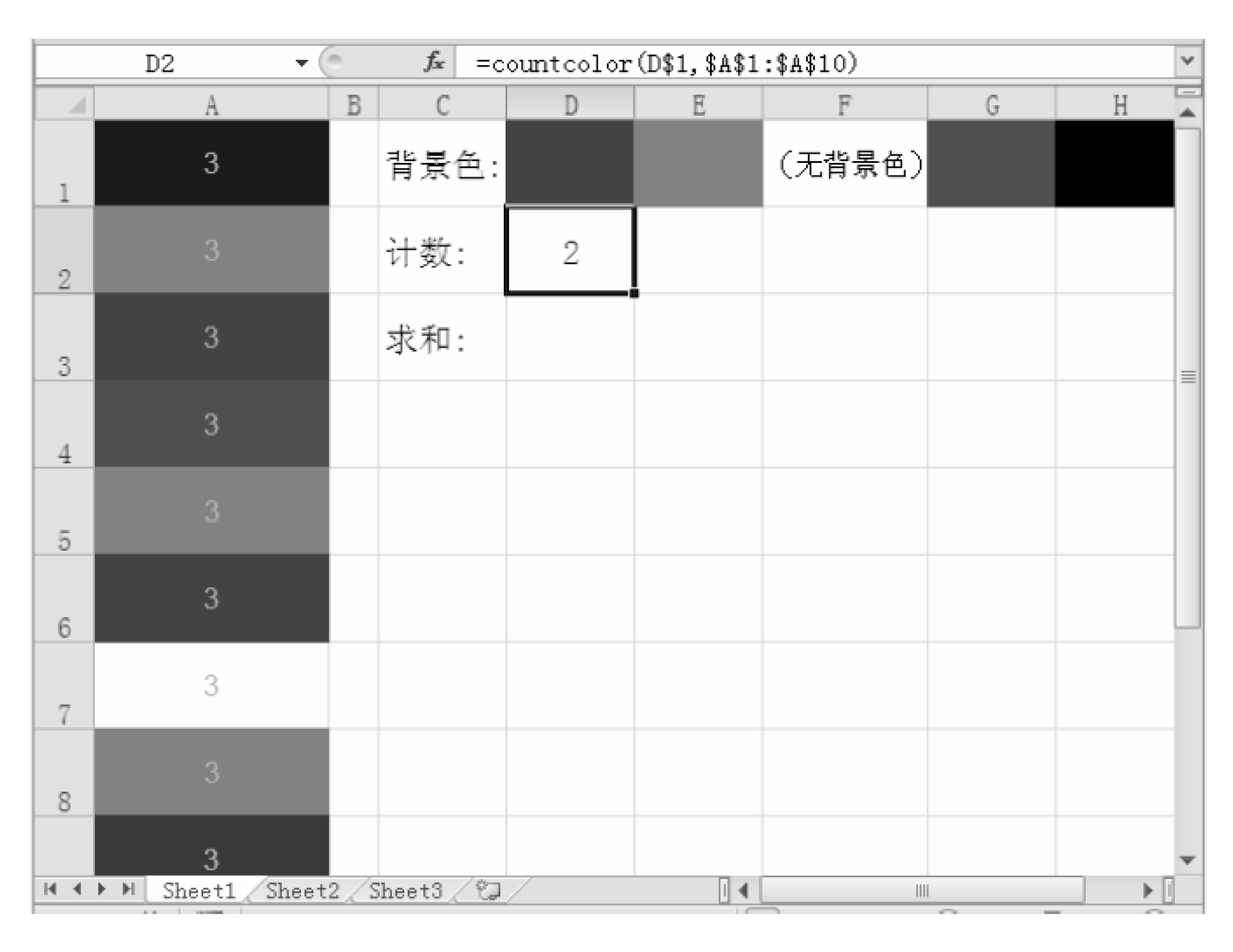


图 13-37　统计出背景颜色为灰色的单元格个数

05 选择 D3 单元格，并输入公式“=sumcolor（D$1,$A$1:$A$10）”，如图 13-38 所示。



图 13-38　输入求和公式

06 按 Enter 键，即可计算出 A1:A10 单元格区域中背景颜色与 D1 单元格中的背景颜色相同的单元格中数值的和，如图 13-39 所示。

D3 =sumcolor(D$1,$A$1:$A$10)

	A	B	C	D	E	F	G	H
1	3		背景色:			(无背景色)		
2	3		计数:	2				
3	3		求和:	6				
4	3							
5	3							
6	3							
7	3							
8	3							
9	3							

Sheet1 Sheet2 Sheet3

图 13-39　计算出所有灰色单元格中数值的和

07 按照相同的方法，统计并计算其他单元格中的背景色与 A1:A10 单元格区域内背景颜色相同的个数以及相同背景颜色的单元格的数值的和，如图 13-40 所示。


D2 =countcolor(D$1,$A$1:$A$10)

	A	B	C	D	E	F	G	H	I
1	3		背景色:			(无背景色)			
2	3		计数:	2	3	1	1	1	
3	3		求和:	6	9	3	3	3	
4	3								
5	3								
6	3								
7	3								
8	3								
9	3								
10	3								

Sheet1 Sheet2 Sheet3

图 13-40　统计并计算其他单元格

通过上面的几个典型的实例讲述了 Excel 2010 中自定义函数使用的全过程，如果用户在工作需要自定义函数，想进一步提高使用自定义函数的水平，则需要注意以下几点：

（1）熟练地掌握 Excel 2010 内置的函数。能用内置函数妥善解决的问题，就不必使用自定义函数。实际上，自定义函数的执行效率比 Excel 2010 内置函数的慢。

（2）认真掌握好 VBA 的基础知识。这点很容易理解，如果连 VBA 的基本规则都不清楚，别说是写出精致的自定义函数，就是写出能解决问题的自定义函数也还大有疑问。

（3）具体写自定义函数代码之前，应该认真分析自己要处理的实际问题，如果这个问题有实际的数学函数模型，那么最好列出这个函数的解析式。



13.7 高手私房菜

技巧 1：将阿拉伯数字快速转换成大写形式

在财务统计时，为了数据安全，往往会将阿拉伯数字表示为人民币大写形式，但是依靠人工来实现根本不现实。尽管在 Excel 2010 中可以利用单元格格式把数字显示为中文大写数字，但是这样并不符合人民币的说明习惯，而要利用现有的工作表内置函数进行转换计算，转换过程又会较为复杂。

编写自定义函数的代码如下。

```
Function N2RMB(M)
    y = Int(Round(100 * Abs(M)) / 100)
    j = Round(100 * Abs(M) + 0.00001) - y * 100
    f = (j / 10 - Int(j / 10)) * 10
    A = IIf(y < 1, "", Application.Text(y, "[DBNum2]") & "元")
    b = IIf(j > 9.5, Application.Text(Int(j / 10), "[DBNum2]") & "角", IIf(y
< 1, "", IIf(f > 1, "零", "")))
    c = IIf(f < 1, "整", Application.Text(Round(f, 0), "[DBNum2]") & "分")
    N2RMB = IIf(Abs(M) < 0.005, "", IIf(M < 0, "负" & A & b & c, A & b & c))
End Function
```

技巧 2：理解自定义函数的误区

在使用自定义函数时存在以下两个误区。

（1）编写完成的自定义函数，在其他工作簿使用时，函数返回的是错误符号“#NAME?”。

自定义函数一般情况下只能在含有函数代码的工作簿内使用。如果需要让该自定义函数在所有打开的 Excel 2010 工作簿中使用，需要保存为加载宏文件并进行加载。

（2）利用函数在单元格中返回引用单元格的格式，如字体、大小、颜色等。

在工作表中使用的函数只能返回值，而不能改变工作表、单元格等内容的结构。

第 14 章　用 VBA 实现数据分析

利用 Excel VBA 程序代码可以进行复杂的数据分析，这在现实工作中使用非常广泛。本章以商品销售数据为例，演示使用 Excel VBA 程序代码进行数据分析的操作方法。通过本章的学习，可以使读者轻松掌握完成复杂数据分析任务的技能。



14.1 销售数据分析

一般销售型企业往往会有大量的客户订单与商品库存信息要处理。合理有效地处理这些数据可以准确计算商品的实际需求量，便于补给缺货，同时还可以计算出销售额等信息。本节将主要介绍销售数据的分析方法。

14.1.1 商品库存需求分析

商品销售时需要有充足、合理的库存，过多的库存会增加不必要的投资，库存太少又不能满足销售的需求，所以要合理调整库存。商品库存需求量即订购量与库存量之间的差值，反映了商品的一种供求关系。通过商品库存需求量分析，不仅可以清楚该商品的库存量，掌握库存量与市场订购量之间的关系，而且还可以制订更好的生产计划。

【例 14.1】制作 2012 年第一季度商品库存需求分析表，通过计算各种不同月份的“需求量”来分析库存量与订购量之间的关系。具体的操作步骤如下。

01 新建一个工作簿，在 Sheet1 工作表中，选择 A1:H1 单元格区域，单击功能区【开始】选项卡下的【合并后居中】按钮，将所选单元格合并，并在其中输入“2012 年第一季度商品需求量及销售情况分析”，如图 14-1 所示。

A1　fx　2012年第一季度商品需求量及销售情况分析

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2								
3								
4								
5								
6								
7								

Sheet1　Sheet2　Sheet3

图 14-1　合并单元格并输入标题文字

02 右击合并后的单元格，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，打开【设置单元格格式】对话框。在【字体】选项卡的【字体】列表框中选择"华文新魏"，在【字形】列表框中选择"常规"，在【字号】列表框中选择"16"，在【颜色】下拉列表框中选择要设置的颜色，如图 14-2 所示。

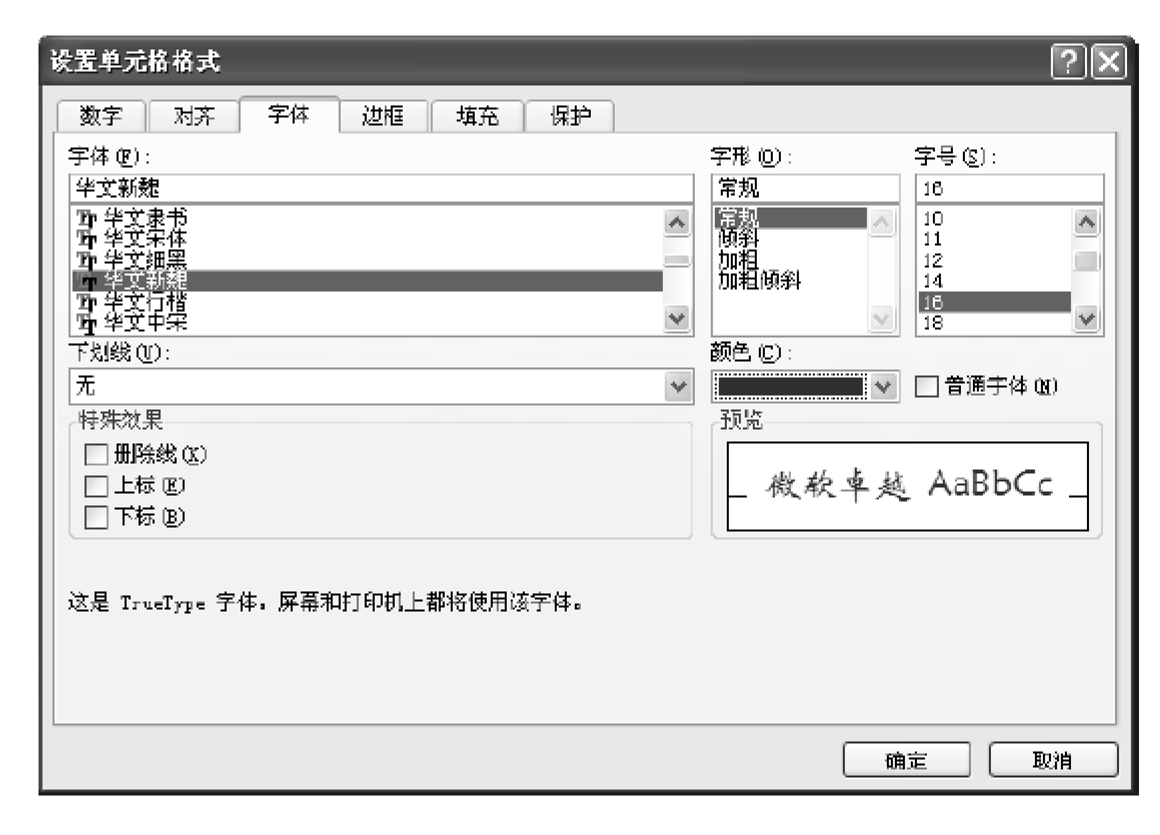


图 14-2　【设置单元格格式】对话框

03 选择【填充】选项卡，为所选单元格设置背景颜色，这里选择浅灰色，如图 14-3 所示。



图 14-3　【填充】选项卡

04 设置完成后，单击【确定】按钮，返回工作表界面，可看到设置后的单元格字体和填充效果，如图 14-4 所示。

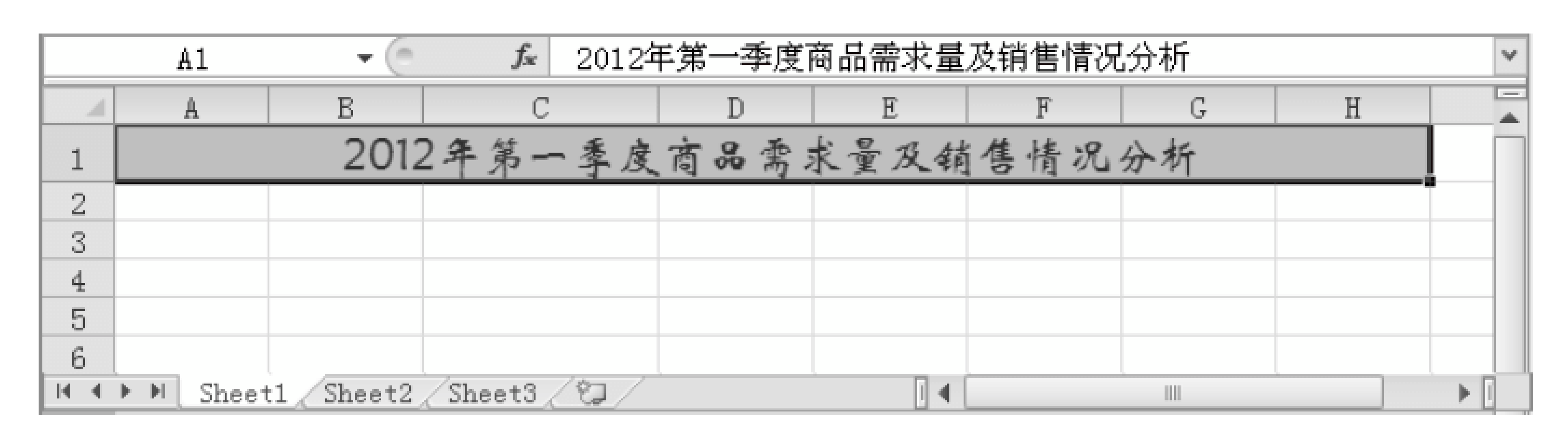


图 14-4　设置的单元格效果

05 在 A2:H2 单元格区域中分别输入"日期"、"品牌"、"产品型号"、"订购量"、"库存量"、"需求量"、"单价"和"销售金额"各项标题，如图 14-5 所示。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3								
4								
5								
6								
7								
8								
9								

图 14-5　输入各标题项

06 右击选择的 A2:H2 单元格区域，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，打开【设置单元格格式】对话框，在其中设置各标题的字体为"宋体"，字号为"12"，并设置单元格区域的背景颜色为浅蓝色，如图 14-6 所示。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3								
4								
5								
6								
7								

图 14-6　设置字体及单元格背景色

07 在 A3:G29 单元格区域中，输入如图 14-7 所示的各项数据信息。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200		￥3,799	
4			Y470N-IFI(H)	120	100		￥5,500	
5			ThinkPad E420	150	80		￥4,799	
6		惠普	G4-1207tx	60	120		￥3,849	
7			g4-1012TX	80	120		￥4,099	
8			g4-1058TX	100	100		￥3,800	
9		苹果	Pro(MC700CH/A)	40	50		￥7,686	
10			Air(MC968CH/A)	70	45		￥7,398	
11			Air(MC965CH/A)	60	40		￥9,598	
12	2月	联想	G470AH-ITH	120	80		￥3,799	
13			Y470N-IFI(H)	100	120		￥5,500	
14			ThinkPad E420	130	100		￥4,799	
15		惠普	G4-1207tx	130	90		￥3,849	
16			g4-1012TX	90	50		￥4,099	
17			g4-1058TX	80	55		￥3,800	
18		苹果	Pro(MC700CH/A)	50	100		￥7,686	
19			Air(MC968CH/A)	80	64		￥7,398	
20			Air(MC965CH/A)	60	80		￥9,598	
21	3月	联想	G470AH-ITH	200	165		￥3,799	
22			Y470N-IFI(H)	180	126		￥5,500	
23			ThinkPad E420	110	96		￥4,799	

Sheet1　Sheet2　Sheet3

图 14-7　输入各项数据信息

08 输入完毕后，右击选择的 A1:H29 单元格区域，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，打开【设置单元格格式】对话框，选择【边框】选项卡，从中为所选单元格区域设置单元格的边框颜色及样式，如图 14-8 所示。



图 14-8　【边框】选项卡

09 设置完成后，单击【确定】按钮，返回工作表，可看到设置边框后的效果，如图 14-9 所示。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200		￥3,799	
4			Y470N-IFI(H)	120	100		￥5,500	
5			ThinkPad E420	150	80		￥4,799	
6		惠普	G4-1207tx	60	120		￥3,849	
7			g4-1012TX	80	120		￥4,099	
8			g4-1058TX	100	100		￥3,800	
9		苹果	Pro(MC700CH/A)	40	50		￥7,686	
10			Air(MC968CH/A)	70	45		￥7,398	
11			Air(MC965CH/A)	60	40		￥9,598	
12	2月	联想	G470AH-ITH	120	80		￥3,799	
13			Y470N-IFI(H)	100	120		￥5,500	
14			ThinkPad E420	130	100		￥4,799	
15		惠普	G4-1207tx	130	90		￥3,849	
16			g4-1012TX	90	50		￥4,099	
17			g4-1058TX	80	55		￥3,800	
18		苹果	Pro(MC700CH/A)	50	100		￥7,686	
19			Air(MC968CH/A)	80	64		￥7,398	
20			Air(MC965CH/A)	60	80		￥9,598	
21	3月	联想	G470AH-ITH	200	165		￥3,799	
22			Y470N-IFI(H)	180	126		￥5,500	
23			ThinkPad E420	110	96		￥4,799	
24		惠普	G4-1207tx	150	200		￥3,849	

Sheet1　Sheet2　Sheet3

图 14-9　设置单元格边框属性

10 选择工作表中功能区的【开发工具】选项卡，单击【代码】选项组中的【Visual Basic】按钮，打开【Microsoft Visual Basic for Application】主窗口，在菜单栏选择【插入】➢【模块】菜单命令，插入一个模块，并命名为“模块 1”。双击进入该模块代码窗口中，然后在菜单栏选择【插入】➢【过程】菜单命令，打开【添加过程】对话框，在【名称】文本框中输入“统计需求量”，在【类型】选项组中选中【子过程】单选按钮，在【范围】选项组中选中【公共的】单选按钮，如图 14-10 所示。

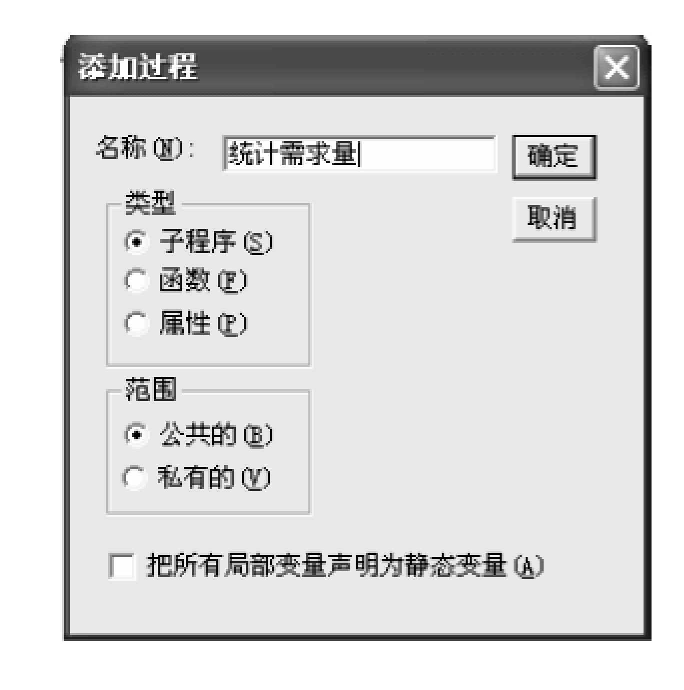


图 14-10　【添加过程】对话框

11 单击【确定】按钮，在模块 1 中添加一个子过程的代码框架，如图 14-11 所示。

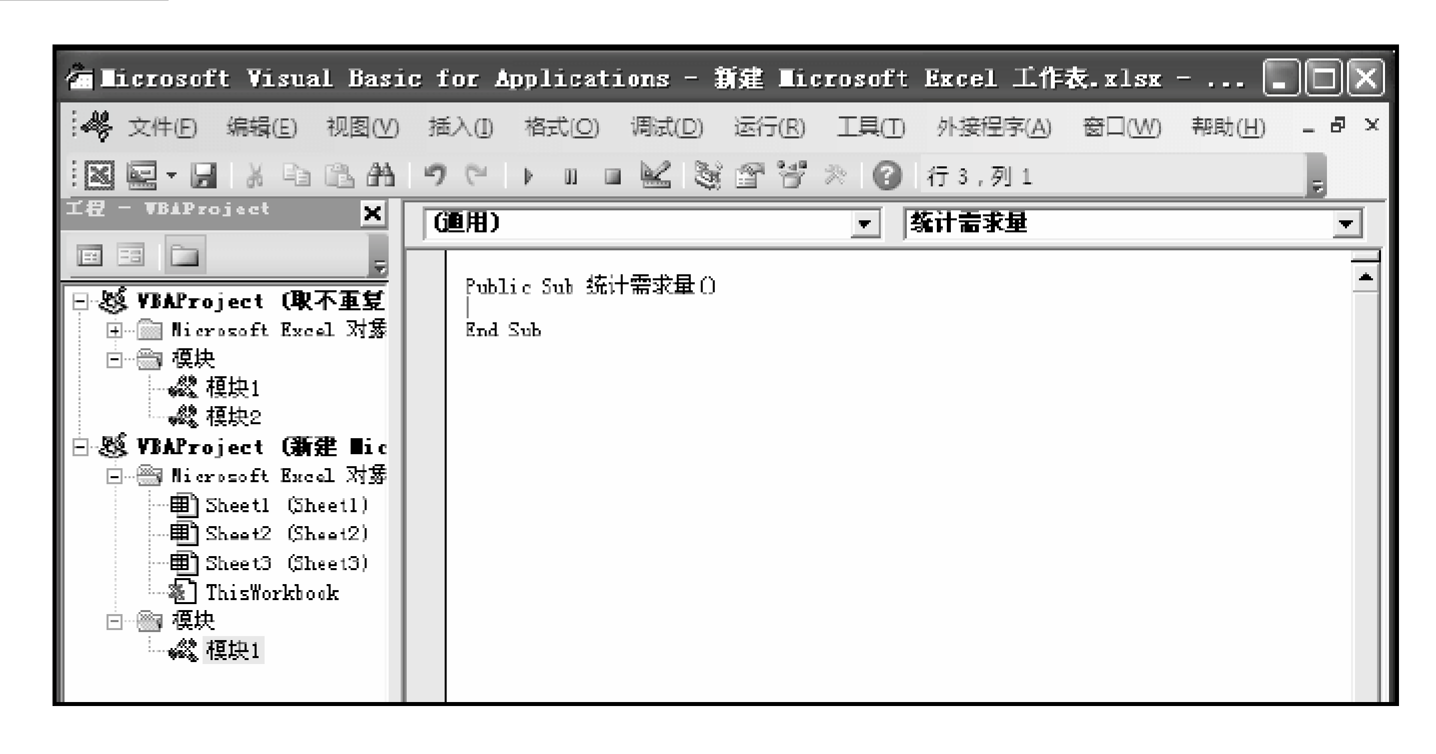


图 14-11　添加子过程的代码框架

12 在该子过程的代码框架中输入以下程序代码，如图 14-12 所示。

```
Public Sub 统计需求量()
Dim i As Integer    '定义用于循环的整型变量
   For i = 3 To 29
       Cells(i, 6).Select
       If Cells(i, 4) > Cells(i, 5) Then
             '判断订购量与库存量之间的大小关系,如果订购量大于库存量
           Cells(i, 6) = Cells(i, 4) - Cells(i, 5)  '需求量=订购量-库存量
           Selection.Interior.ColorIndex = 15  '设置单元格的底色为“灰色-25%”
       Else
            Cells(i, 6) = "无"
                 With Selection
                   .Interior.ColorIndex = 37 '设置单元格填充颜色为“淡蓝色”
                   .HorizontalAlignment = xlCenter  '设置居中对齐
                   .VerticalAlignment = xlCenter
                   .Font.Name = "Times New Roman"  '设置字体格式
                   .Font.Size = 12
                   .Font.Bold = True
                   .Font.ColorIndex = 3
                 End With
       End If
   Next i
End Sub
```


(通用)　统计需求量

```
Public Sub 统计需求量()
Dim i As Integer    '定义用于循环的整型变量
   For i = 3 To 29
       Cells(i, 6).Select
       If Cells(i, 4) > Cells(i, 5) Then
             '判断订购量与库存量之间的大小关系,如果订购量大于库存量
           Cells(i, 6) = Cells(i, 4) - Cells(i, 5)  '需求量=订购量-库存量
           Selection.Interior.ColorIndex = 15  '设置单元格的底色为"灰色-25%"
       Else
            Cells(i, 6) = "无"
                 With Selection
                   .Interior.ColorIndex = 37 '设置单元格填充颜色为"淡蓝色"
                   .HorizontalAlignment = xlCenter  '设置居中对齐
                   .VerticalAlignment = xlCenter
                   .Font.Name = "Times New Roman"  '设置字体格式
                   .Font.Size = 12
                   .Font.Bold = True
                   .Font.ColorIndex = 3
                 End With
       End If
   Next i
End Sub
```


图 14-12　输入完整的程序代码

【提示】这段代码首先利用 For 循环在第 3 行到第 29 行单元格区域内逐行比较商品订购量与库存量之间的大小关系，然后根据判断结果来计算商品的需求量，并对需求量所在单元格进行相应的格式设置。

13 为了方便程序代码的执行，可以在工作表中为其设置执行按钮。选择功能区【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中选择【按钮】按钮，然后在工作表的适当位置，当鼠标指针变为“+”状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，系统则弹出【指定宏】对话框，在【宏名】列表框中选择“统计需求量”选项，如图 14-13 所示。



图 14-13　【指定宏】对话框

14 单击【确定】按钮，完成宏的指定操作。右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为“计算需求量”，然后单击工作表中的任意单元格，退出该按钮的编辑状态，如图 14-14 所示。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200		￥3,799	
4			Y470N-IFI(H)	120	100		￥5,500	
5			ThinkPad E420	150	80		￥4,799	
6		惠普	G4-1207tx	60	120		￥3,849	
7			g4-1012TX	80	120		￥4,099	
8			g4-1058TX	100	100		￥3,800	
9		苹果	Pro(MC700CH/A)	40	50		￥7,686	
10			Air(MC968CH/A)	70	45		￥7,398	
11			Air(MC965CH/A)	60	40		￥9,598	
12	2月	联想	G470AH-ITH	120	80		￥3,799	
13			Y470N-IFI(H)	100	120		￥5,500	
14			ThinkPad E420	130	100		￥4,799	
15		惠普	G4-1207tx	130	90		￥3,849	
16			g4-1012TX	90	50		￥4,099	
17			g4-1058TX	80	55		￥3,800	
18		苹果	Pro(MC700CH/A)	50	100		￥7,686	
19			Air(MC968CH/A)	80	64		￥7,398	
20			Air(MC965CH/A)	60	80		￥9,598	
21	3月	联想	G470AH-ITH	200	165		￥3,799	
22			Y470N-IFI(H)	180	126		￥5,500	
23			ThinkPad E420	110	96		￥4,799	
24		惠普	G4-1207tx	150	200		￥3,849	

计算需求量

Sheet1 Sheet2 Sheet3

图 14-14　更改按钮名称

15 单击【计算需求量】按钮，执行指定的过程，得到的结果如图 14-15 所示。

	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200	无	￥3,799	
4			Y470N-IFI(H)	120	100	20	￥5,500	
5			ThinkPad E420	150	80	70	￥4,799	
6		惠普	G4-1207tx	60	120	无	￥3,849	
7			g4-1012TX	80	120	无	￥4,099	
8			g4-1058TX	100	100	无	￥3,800	
9		苹果	Pro(MC700CH/A)	40	50	无	￥7,686	
10			Air(MC968CH/A)	70	45	25	￥7,398	
11			Air(MC965CH/A)	60	40	20	￥9,598	
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	
13			Y470N-IFI(H)	100	120	无	￥5,500	
14			ThinkPad E420	130	100	30	￥4,799	
15		惠普	G4-1207tx	130	90	40	￥3,849	
16			g4-1012TX	90	50	40	￥4,099	
17			g4-1058TX	80	55	25	￥3,800	
18		苹果	Pro(MC700CH/A)	50	100	无	￥7,686	
19			Air(MC968CH/A)	80	64	16	￥7,398	
20			Air(MC965CH/A)	60	80	无	￥9,598	
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	
22			Y470N-IFI(H)	180	126	54	￥5,500	
23			ThinkPad E420	110	96	14	￥4,799	
24		惠普	G4-1207tx	150	200	无	￥3,849	

计算需求量

Sheet1 Sheet2 Sheet3

图 14-15　计算出需求量

16 可在上述代码窗口中添加下面的代码，将库存充足的商品记录隐藏起来，这样可以突出显示库存需求量不足的商品，如图 14-16 所示。

(通用)　　隐藏

```
Public Sub 统计需求量()
Dim i As Integer      '定义用于循环的整型变量
   For i = 3 To 29
       Cells(i, 6).Select
       If Cells(i, 4) > Cells(i, 5) Then
            '判断订购量与库存量之间的大小关系，如果订购量大于库存量
          Cells(i, 6) = Cells(i, 4) - Cells(i, 5)  '需求量=订购量-库存量
          Selection.Interior.ColorIndex = 15   '设置单元格的底色为"灰色-25%"
       Else
          Cells(i, 6) = "无"
             With Selection
              .Interior.ColorIndex = 37 '设置单元格填充颜色为"淡蓝色"
              .HorizontalAlignment = xlCenter  '设置居中对齐
              .VerticalAlignment = xlCenter
              .Font.Name = "Times New Roman"  '设置字体格式
              .Font.Size = 12
              .Font.Bold = True
              .Font.ColorIndex = 3
             End With
       End If
   Next i
End Sub

Public Sub 隐藏()
Dim i As Integer      '定义用于循环的整型变量
   For i = 3 To 29
     If Cells(i, 6) = "无" Then
        Rows(i).Select   '选择第i行
        Selection.EntireRow.Hidden = True  '隐藏第i行
     End If
   Next i
End Sub
```

图 14-16　输入隐藏充足商品记录的代码

```
Public Sub 隐藏()
Dim i As Integer     '定义用于循环的整型变量
   For i = 3 To 29
     If Cells(i, 6) = "无" Then
        Rows(i).Select  '选择第 i 行
        Selection.EntireRow.Hidden = True  '隐藏第 i 行
     End If
   Next i
End Sub
```

17 关闭 VBA 代码窗口，返回工作簿中，选择功能区【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中选择【按钮】按钮 ▬，然后在工作表的适当位置，当鼠标指针变为

“+”状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，系统则弹出【指定宏】对话框，在【宏名】列表框中选择“隐藏”选项，如图 14-17 所示。



图 14-17　【指定宏】对话框

18 单击【确定】按钮，完成宏的指定操作。然后右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为“隐藏库存充足产品”，然后单击工作表中的任意单元格，退出该按钮的编辑状态，如图 14-18 所示。

	A	B	C	D	E	F	G	H
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200	无	￥3,799	
4			Y470N-IFI(H)	120	100	20	￥5,500	
5			ThinkPad E420	150	80	70	￥4,799	
6		惠普	G4-1207tx	60	120	无	￥3,849	
7			g4-1012TX	80	120	无	￥4,099	
8			g4-1058TX	100	100	无	￥3,800	
9		苹果	Pro(MC700CH/A)	40	50	无	￥7,686	
10			Air(MC968CH/A)	70	45	25	￥7,398	
11			Air(MC965CH/A)	60	40	20	￥9,598	
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	
13			Y470N-IFI(H)	100	120	无	￥5,500	
14			ThinkPad E420	130	100	30	￥4,799	
15		惠普	G4-1207tx	130	90	40	￥3,849	
16			g4-1012TX	90	50	40	￥4,099	
17			g4-1058TX	80	55	25	￥3,800	
18		苹果	Pro(MC700CH/A)	50	100	无	￥7,686	
19			Air(MC968CH/A)	80	64	16	￥7,398	
20			Air(MC965CH/A)	60	80	无	￥9,598	
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	
22			Y470N-IFI(H)	180	126	54	￥5,500	
23			ThinkPad E420	110	96	14	￥4,799	
24		惠普	G4-1207tx	150	200	无	￥3,849	

计算需求量
隐藏库存充足产品

图 14-18　更改按钮名称

19 单击【隐藏库存充足产品】按钮，执行指定的过程，隐藏库存充足的商品记录，如图 14-19 所示。

	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
4			Y470N-IFI(H)	120	100	20	￥5,500	
5			ThinkPad E420	150	80	70	￥4,799	
10			Air(MC968CH/A)	70	45	25	￥7,398	
11			Air(MC965CH/A)	60	40	20	￥9,598	
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	
14			ThinkPad E420	130	100	30	￥4,799	
15		惠普	G4-1207tx	130	90	40	￥3,849	
16			g4-1012TX	90	50	40	￥4,099	
17			g4-1058TX	80	55	25	￥3,800	
19			Air(MC968CH/A)	80	64	16	￥7,398	
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	
22			Y470N-IFI(H)	180	126	54	￥5,500	
23			ThinkPad E420	110	96	14	￥4,799	
26			g4-1058TX	110	60	50	￥3,800	
27		苹果	Pro(MC700CH/A)	110	85	25	￥7,686	
29			Air(MC965CH/A)	120	68	52	￥9,598	
30								

计算需求量
隐藏库存充足产品

图 14-19　隐藏库存充足行的效果

20 当用户想要查看所有商品的库存需求记录，需要取消隐藏操作，可在 VBA 代码窗口中输入以下取消隐藏充足库存行的程序代码，如图 14-20 所示。

```
Public Sub 取消隐藏()
Dim k As Integer    '定义用于循环的整型变量
   For k = 3 To 29
     If Cells(k, 6) = "无" Then
         Rows(k).Select  '选择第 k 行
         Selection.EntireRow.Hidden = False  '取消该行的隐藏
     End If
   Next k
End Sub
```

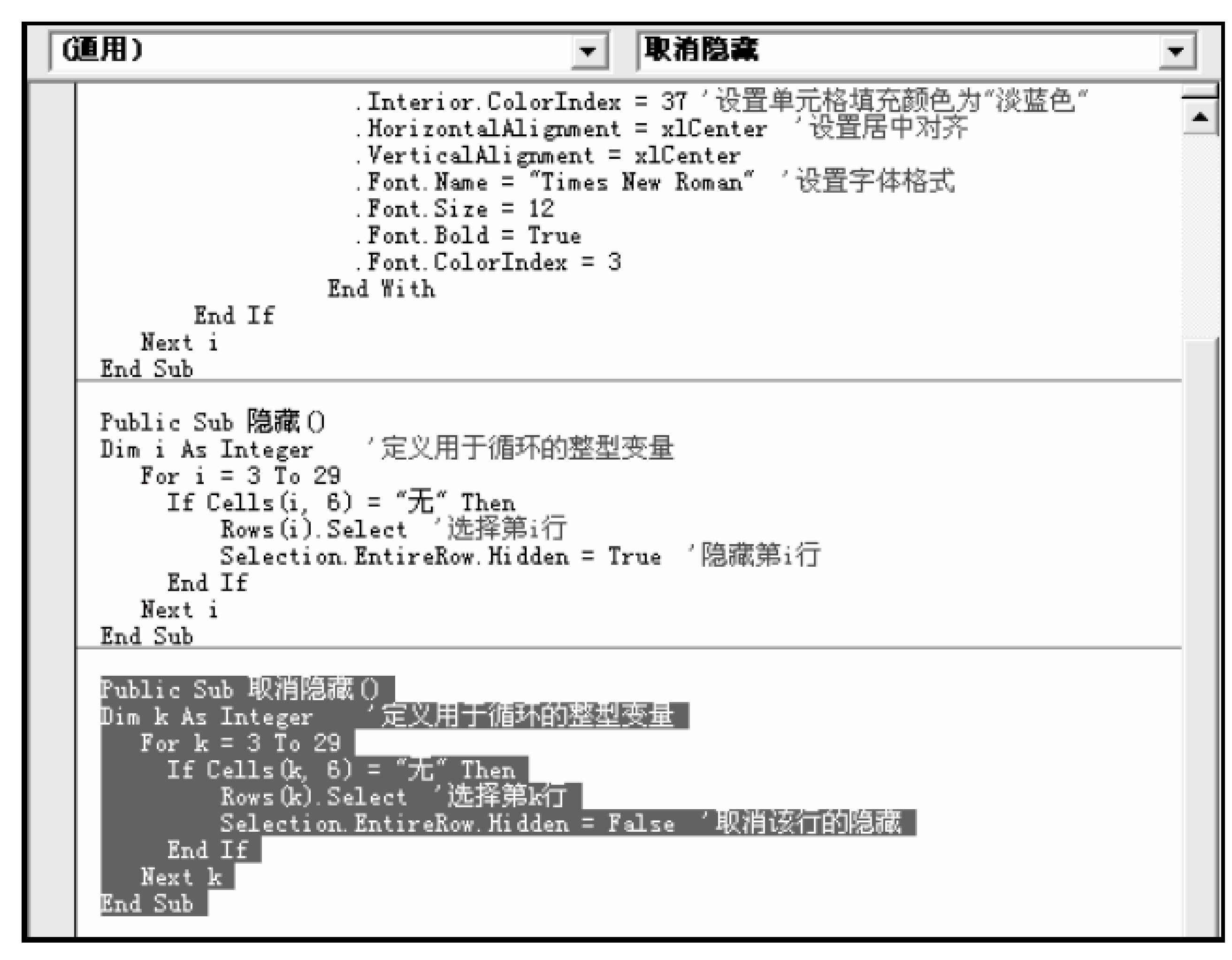


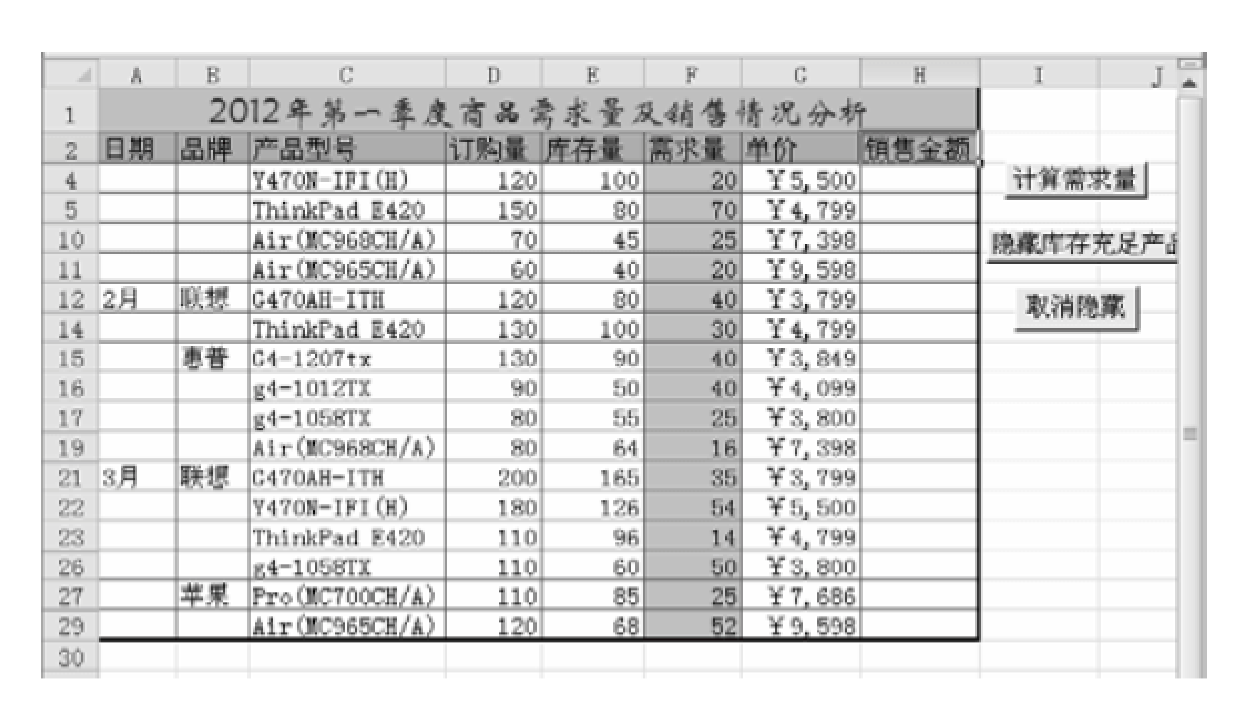
图 14-20 取消隐藏库存充足行的代码

21 按照上文的方法为该过程添加执行按钮。在打开的【指定宏】对话框中选择“取消隐藏”选项，如图 14-21 所示。

22 将按钮名称更改为“取消隐藏”，然后单击工作表中的任意位置退出该按钮的编辑状态，如图 14-22 所示。



图 14-21 【指定宏】对话框



	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
4			Y470N-IFI(H)	120	100	20	￥5,500	
5			ThinkPad E420	150	80	70	￥4,799	
10			Air(MC968CH/A)	70	45	25	￥7,398	
11			Air(MC965CH/A)	60	40	20	￥9,598	
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	
14			ThinkPad E420	130	100	30	￥4,799	
15		惠普	G4-1207tx	130	90	40	￥3,849	
16			g4-1012TX	90	50	40	￥4,099	
17			g4-1058TX	80	55	25	￥3,800	
19			Air(MC968CH/A)	80	64	16	￥7,398	
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	
22			Y470N-IFI(H)	180	126	54	￥5,500	
23			ThinkPad E420	110	96	14	￥4,799	
26			g4-1058TX	110	60	50	￥3,800	
27		苹果	Pro(MC700CH/A)	110	85	25	￥7,686	
29			Air(MC965CH/A)	120	68	52	￥9,598	
30								

图 14-22 命名【取消隐藏】按钮

23 单击【取消隐藏】按钮，即可查看所有商品的记录。

到此，就完成了对商品需求量的分析操作。用户可根据自己的实际情况，灵活地运用 VBA 设计、编辑符合自己工作需要的 VBA 过程。

14.1.2 商品销售情况分析

商品销售情况分析是合理调整销售策略的前提，如果不能准确地掌握销售情况，会影响产品收益。以“2012 年第一季度商品需求量及销售情况分析”表为例，使用 Excel VBA 定期进行商品销售情况分析的方法介绍如下。

1. 计算销售金额

使用 Excel VBA 计算商品销售金额具体操作步骤如下。

01 打开上文中编辑的“2012 年第一季度商品需求量及销售情况分析”工作表 Sheet1，选择功能区【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮，打开 Microsoft Visual Basic for Application 主窗口，在菜单栏选择【插入】➤【模块】菜单命令，插入一个模块，在该模块的【名称】属性列表框中将其更改为“统计销售金额”，如图 14-23 所示。

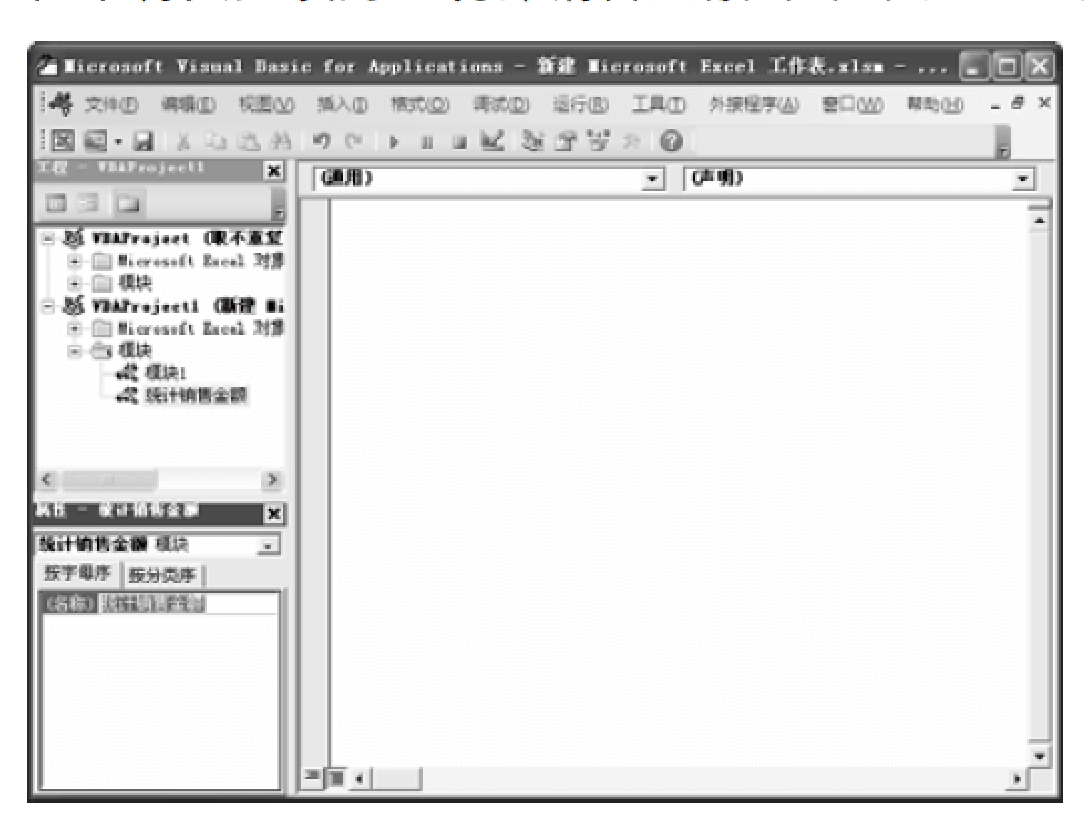


图 14-23 插入模块并更改模块名称

02 双击进入该模块代码窗口中，在其中输入以下实现统计销售金额的代码程序，如图 14-24 所示。

```
Public Sub 分析销售金额()
Dim i As Integer            ' 定义用于循环的整型变量
For i = 3 To 29             ' 设置循环变量
  Cells(i, 8) = Cells(i, 7) * Cells(i, 4)    '销售金额=单价*订购量
Next i
End Sub
```

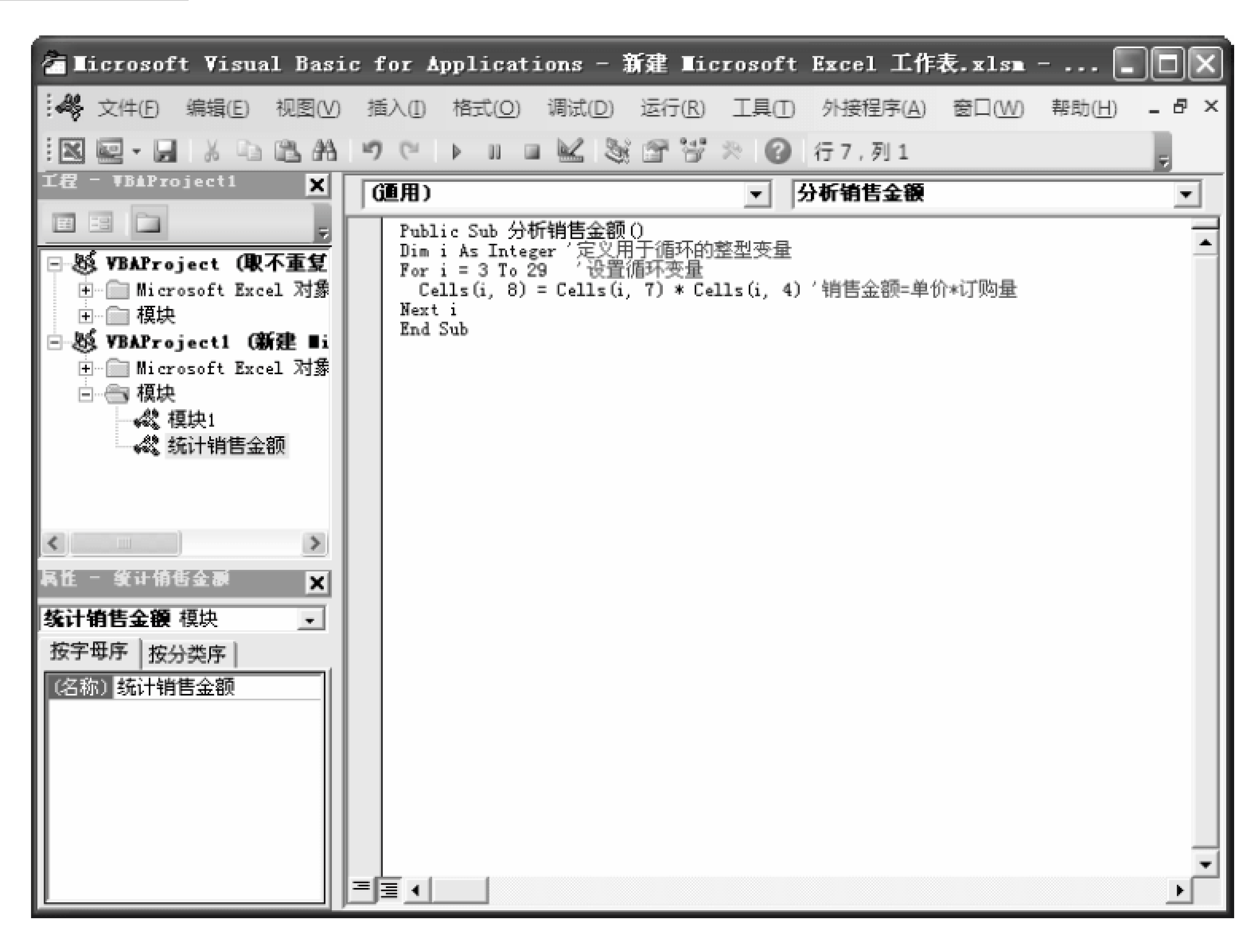


图 14-24　计算销售金额程序代码

03 为了方便程序代码的执行，要在工作表中设置执行按钮。选择功能区【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中选择【按钮】按钮，然后在工作表的适当位置处，当鼠标指针变为"+"状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，系统则弹出【指定宏】对话框，在【宏名】列表中选择"分析销售金额"选项，如图 14-25 所示。



图 14-25　【指定宏】对话框

04 单击【确定】按钮，完成宏的指定操作。然后右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为"计算销售金额"，然后单击工作表中的任意单元格，即可退出该按钮的编辑状态，如图 14-26 所示。

	A	B	C	D	E	F	G	H	I
1	2012年第一季度商品需求量及销售情况分析								
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额	
3	1月	联想	G470AH-ITH	100	200	无	￥3,799		计算需求量
4			Y470N-IFI(H)	120	100	20	￥5,500		
5			ThinkPad E420	150	80	70	￥4,799		
6		惠普	G4-1207tx	60	120	无	￥3,849		隐藏库存充足产品
7			g4-1012TX	80	120	无	￥4,099		
8			g4-1058TX	100	100	无	￥3,800		取消隐藏
9		苹果	Pro(MC700CH/A)	40	50	无	￥7,686		
10			Air(MC968CH/A)	70	45	25	￥7,398		计算销售金额
11			Air(MC965CH/A)	60	40	20	￥9,598		
12	2月	联想	G470AH-ITH	120	80	40	￥3,799		
13			Y470N-IFI(H)	100	120	无	￥5,500		
14			ThinkPad E420	130	100	30	￥4,799		
15		惠普	G4-1207tx	130	90	40	￥3,849		
16			g4-1012TX	90	50	40	￥4,099		
17			g4-1058TX	80	55	25	￥3,800		
18		苹果	Pro(MC700CH/A)	50	100	无	￥7,686		
19			Air(MC968CH/A)	80	64	16	￥7,398		
20			Air(MC965CH/A)	60	80	无	￥9,598		
21	3月	联想	G470AH-ITH	200	165	35	￥3,799		
22			Y470N-IFI(H)	180	126	54	￥5,500		
23			ThinkPad E420	110	96	14	￥4,799		

Sheet1　Sheet2　Sheet3

图 14-26　更改按钮名称

05 单击【计算销售金额】按钮，执行指定的过程，计算出销售金额，显示在 H 列中，如图 14-27 所示。

	A	B	C	D	E	F	G	H	I
1	2012年第一季度商品需求量及销售情况分析								
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额	
3	1月	联想	G470AH-ITH	100	200	无	￥3,799	￥379,900	计算需求量
4			Y470N-IFI(H)	120	100	20	￥5,500	￥660,000	
5			ThinkPad E420	150	80	70	￥4,799	￥719,850	
6		惠普	G4-1207tx	60	120	无	￥3,849	￥230,940	隐藏库存充足产
7			g4-1012TX	80	120	无	￥4,099	￥327,920	
8			g4-1058TX	100	100	无	￥3,800	￥380,000	取消隐藏
9		苹果	Pro(MC700CH/A)	40	50	无	￥7,686	￥307,440	
10			Air(MC968CH/A)	70	45	25	￥7,398	￥517,860	计算销售金额
11			Air(MC965CH/A)	60	40	20	￥9,598	￥575,880	
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	￥455,880	
13			Y470N-IFI(H)	100	120	无	￥5,500	￥550,000	
14			ThinkPad E420	130	100	30	￥4,799	￥623,870	
15		惠普	G4-1207tx	130	90	40	￥3,849	￥500,370	
16			g4-1012TX	90	50	40	￥4,099	￥368,910	
17			g4-1058TX	80	55	25	￥3,800	￥304,000	
18		苹果	Pro(MC700CH/A)	50	100	无	￥7,686	￥384,300	
19			Air(MC968CH/A)	80	64	16	￥7,398	￥591,840	
20			Air(MC965CH/A)	60	80	无	￥9,598	￥575,880	
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	￥759,800	
22			Y470N-IFI(H)	180	126	54	￥5,500	￥990,000	
23			ThinkPad E420	110	96	14	￥4,799	￥527,890	

Sheet1　Sheet2　Sheet3

图 14-27　计算销售金额结果

2. 将计算结果图表化

数字并不能直观地表示统计分析结果，所以可以将其表示为更具表现力的图表形式。将计算得到的结果以图表的形式显示出来的具体操作步骤如下。

01 选择工作表功能区的【开发工具】选项卡，单击【代码】选项组中的【Visual Basic】按钮，打开【Microsoft Visual Basic for Application】主窗口，在菜单栏选择【插入】➢【模块】菜单命令，插入一个模块，在该模块的【名称】属性列表框中将其更改为“销售额图表”，如图 14-28 所示。

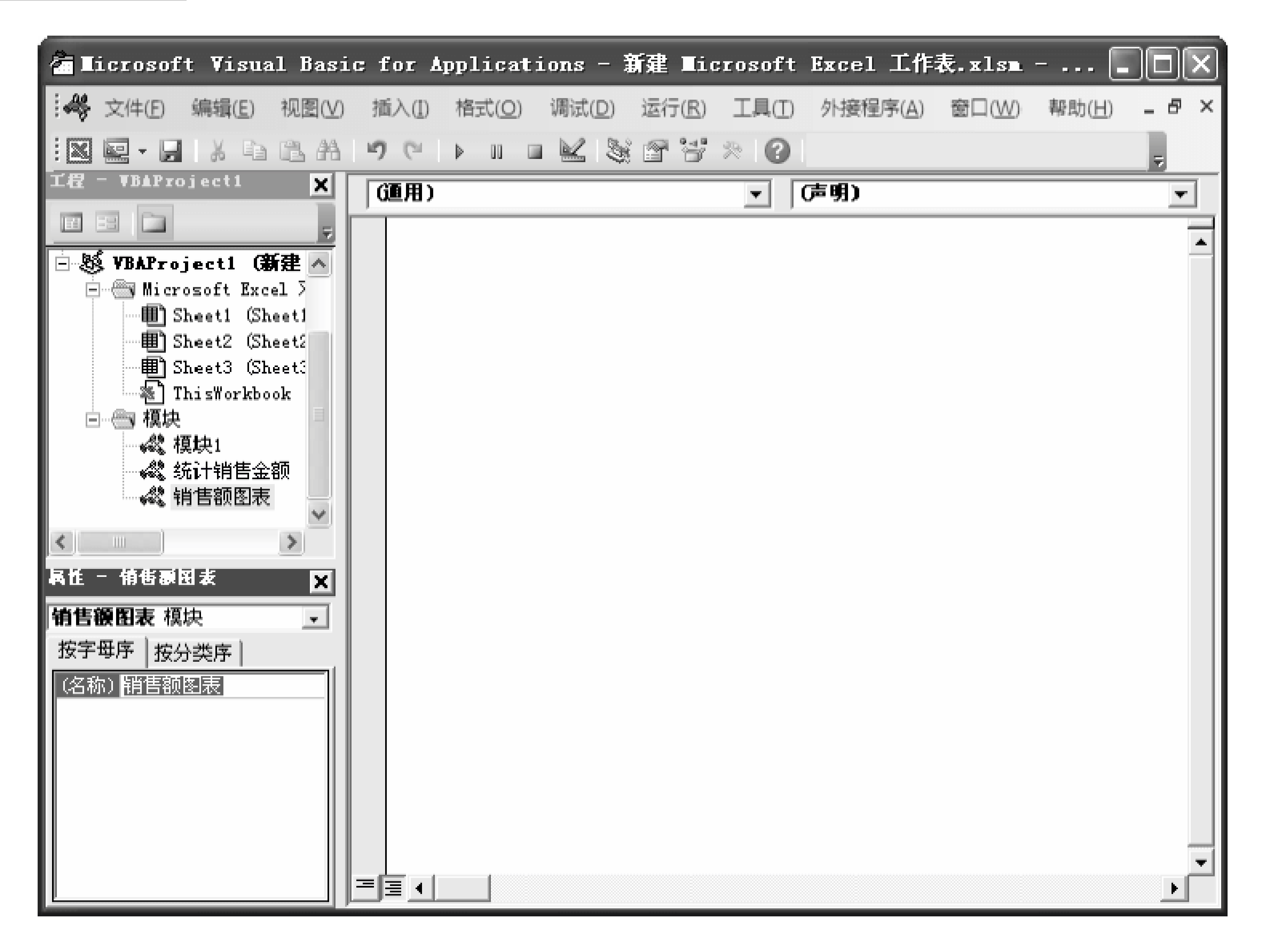


图 14-28　插入模块并更改模块名称

02 双击该模块进入代码窗口，在菜单栏选择【插入】➤【过程】菜单命令，打开【添加过程】对话框，在【名称】文本框中输入"销售额图表"，在【类型】选项组中选中【子过程】单选按钮，在【范围】选项组中选中【公共的】单选按钮，如图 14-29 所示。

03 单击【确定】按钮，在模块中添加了子过程的代码框架，如图 14-30 所示。

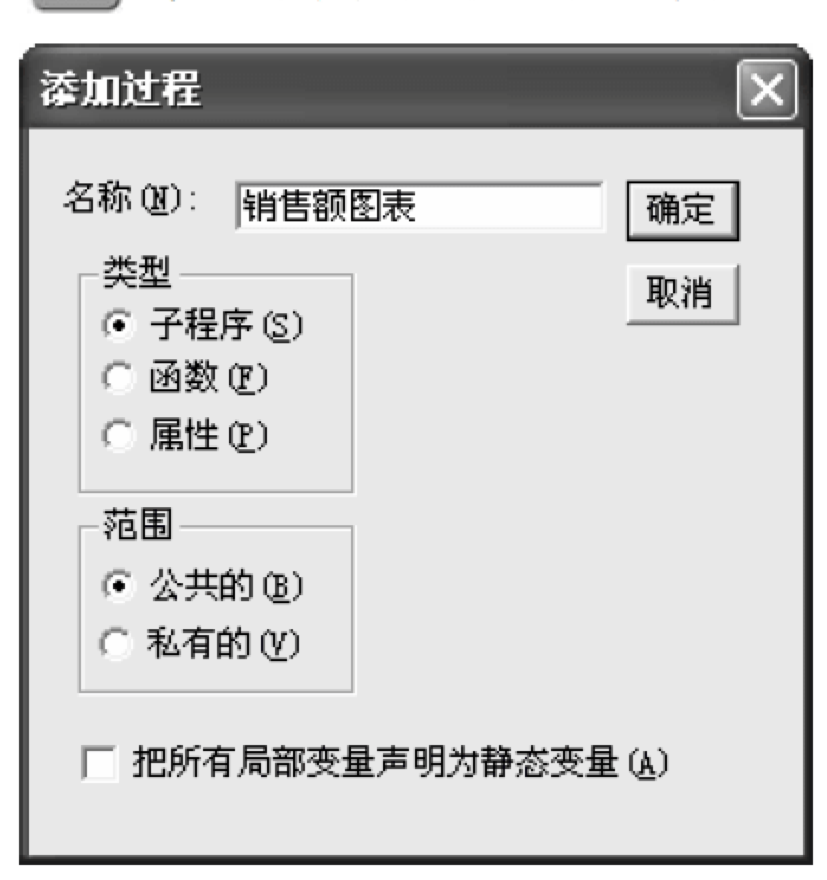


图 14-29　【添加过程】对话框

图 14-30　添加子过程的代码框架

04 在该子过程中输入以下程序代码，如图 14-31 所示。

```
Public Sub 销售额图表 ()
Charts.Add  '添加一个图表
ActiveChart.ChartType = xlLineMarkers     '选择图表类型为“折线图”
ActiveChart.SetSourceData Source:=Sheets("Sheet1").
Range("A2:C29,H2:H29"), PlotBy:=xlColumns   '选择数据来源
ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet1"     '指定添加
图'表的位置
```

```
        With ActiveChart                        '设置图表格式
            .HasTitle = True
            .ChartTitle.Characters.Text = "销售金额分析"      '设置图表标题为"销售金额分
析
            .Axes(xlValue, xlPrimary).HasTitle = True        '设置图表显示 y 轴坐标标题
            .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "销售额"    '设置
'y 轴坐标标题
        End With
    End Sub
```

```
(通用)                                    销售额图表化
Public Sub 销售额图表化()
Charts.Add  '添加一个图表
ActiveChart.ChartType = xlLineMarkers      '选择图表类型为"折线图"
ActiveChart.SetSourceData Source:=Sheets("Sheet1").
Range("A2:C29,H2:H29"), PlotBy:=xlColumns   '选择数据来源
ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet1"     '指定添加图表
   With ActiveChart                          '设置图表格式
       .HasTitle = True
       .ChartTitle.Characters.Text = "销售金额分析"     '设置图表标题为"销售金额
       .Axes(xlValue, xlPrimary).HasTitle = True       '设置图表显示y轴坐标标题
       .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "销售额"     '设置y轴
   End With
End Sub
```

图 14-31　输入完整的程序代码

【提示】代码中，首先使用 Charts.Add 方法创建一个图表对象，并设置其图表类型为“折线图”，然后指定图表的数据源及其显示位置，最后在 With 语句中设置图表的坐标轴情况及其标题内容。

05 按照上文中介绍的方法，为该过程设置执行按钮，并在【指定宏】对话框的【宏名】列表框中选择“销售额图表”选项，如图 14-32 所示。

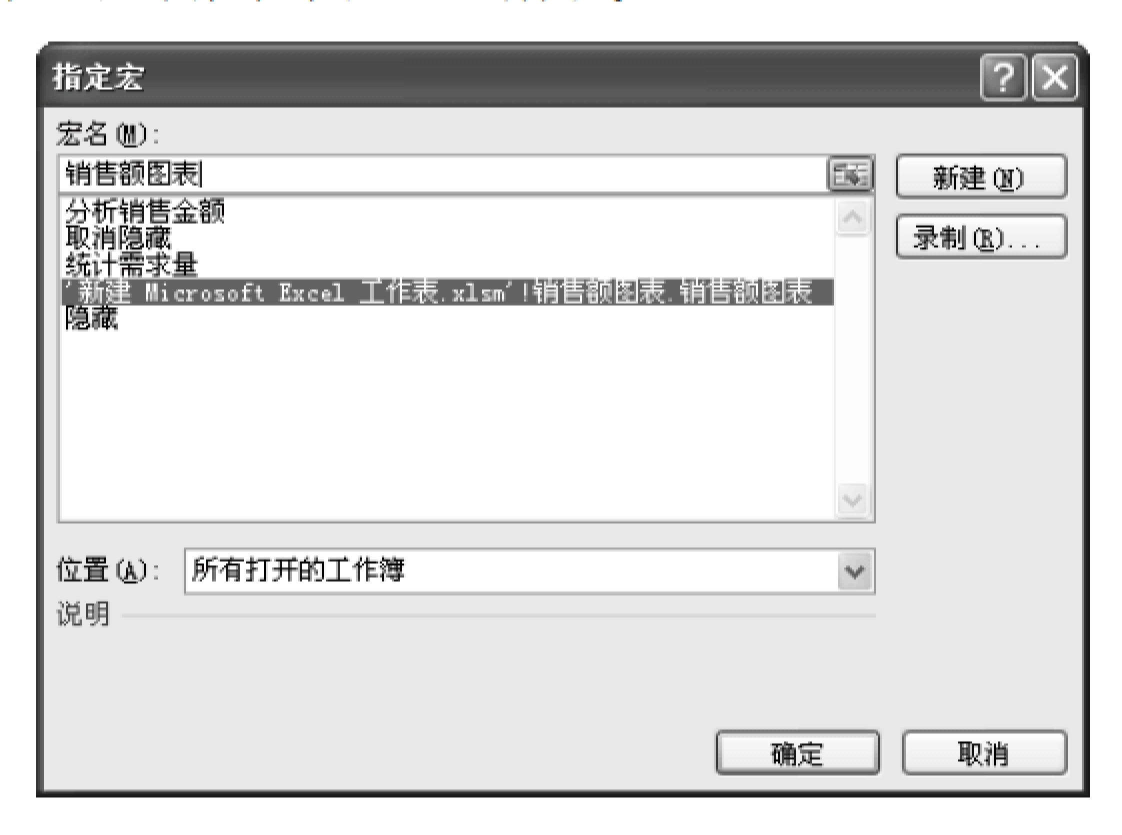


图 14-32　【指定宏】对话框

06 单击【确定】按钮，完成宏的指定操作。然后右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为“销售金额图表”，然后单击工作表中的任意单元格，退出该按钮的编辑状态，如图 14-33 所示。

2012年第一季度商品需求量及销售情况分析						
品牌	产品型号	订购量	库存量	需求量	单价	销售金额
联想	G470AH-ITH	100	200	无	￥3,799	￥379,900
	Y470N-IFI(H)	120	100	20	￥5,500	￥660,000
	ThinkPad E420	150	80	70	￥4,799	￥719,850
惠普	G4-1207tx	60	120	无	￥3,849	￥230,940
	g4-1012TX	80	120	无	￥4,099	￥327,920
	g4-1058TX	100	100	无	￥3,800	￥380,000
苹果	Pro(MC700CH/A)	40	50	无	￥7,686	￥307,440
	Air(MC968CH/A)	70	45	25	￥7,398	￥517,860
	Air(MC965CH/A)	60	40	20	￥9,598	￥575,880
联想	G470AH-ITH	120	80	40	￥3,799	￥455,880
	Y470N-IFI(H)	100	120	无	￥5,500	￥550,000
	ThinkPad E420	130	100	30	￥4,799	￥623,870
惠普	G4-1207tx	130	90	40	￥3,849	￥500,370
	g4-1012TX	90	50	40	￥4,099	￥368,910
	g4-1058TX	80	55	25	￥3,800	￥304,000
苹果	Pro(MC700CH/A)	50	100	无	￥7,686	￥384,300
	Air(MC968CH/A)	80	64	16	￥7,398	￥591,840
	Air(MC965CH/A)	60	80	无	￥9,598	￥575,880
联想	G470AH-ITH	200	165	35	￥3,799	￥759,800
	Y470N-IFI(H)	180	126	54	￥5,500	￥990,000
	ThinkPad E420	110	96	14	￥4,799	￥527,890

计算需求量
隐藏库存充足产品
取消隐藏
计算销售金额
销售金额图表

图 14-33　更改按钮名称

07 单击【销售金额图表】按钮，执行指定的过程，在指定位置创建图表，如图 14-34 所示。

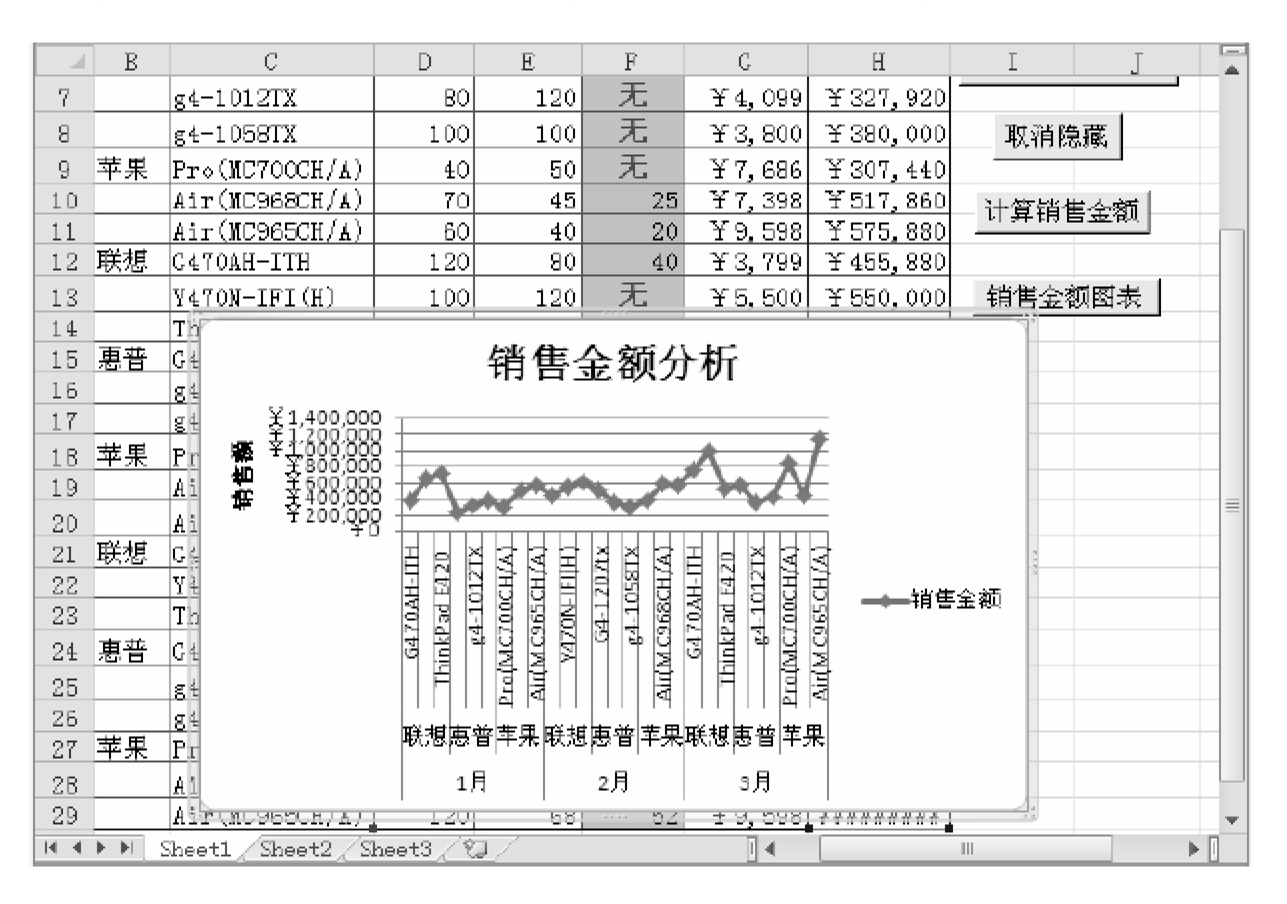


图 14-34　创建图表

08 创建图表后，可以使用鼠标调整图表的大小及位置，也可以设置图表格式。例如，右击选中的图表标题，在弹出的快捷菜单中选择【字体】菜单命令，打开【字体】对话框，在【字体】选项卡的【中文字体】下拉列表框中选择“华文新魏”，在【字体样式】下拉列表框中选择“加粗”，在【大小】下拉列表框中选择“20”，并设置字体颜色为深红色，如图 14-35 所示。



图 14-35　【字体】对话框

09 单击【确定】按钮，即可看到标题字体、颜色和大小已经改变，如图 14-36 所示。

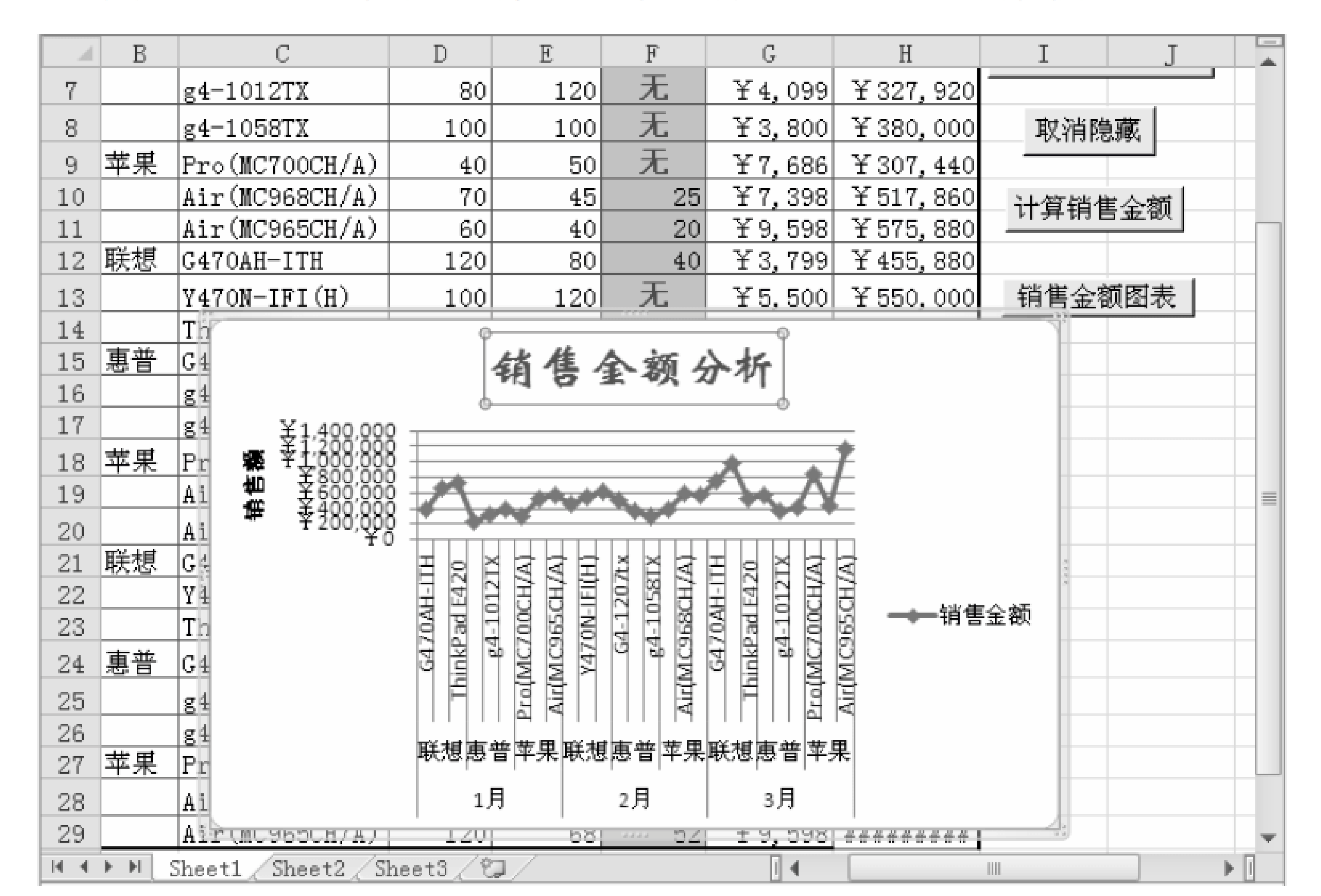


图 14-36　改变标题字体、颜色和大小后的效果

10 除了可以设置字体外，还可以为标题设置填充色、边框颜色、边框样式、阴影等效果。例如，右击选择的图表标题，在弹出的快捷菜单中选择【设置图表标题格式】菜单命令，打开【设置图表标题格式】对话框，选择【填充】选项卡，在其右侧列表中选中【图案填充】单选按钮，在设置前景色和背景色后，在图案填充列表框中选择要设置的图案样式，如图 14-37 所示。

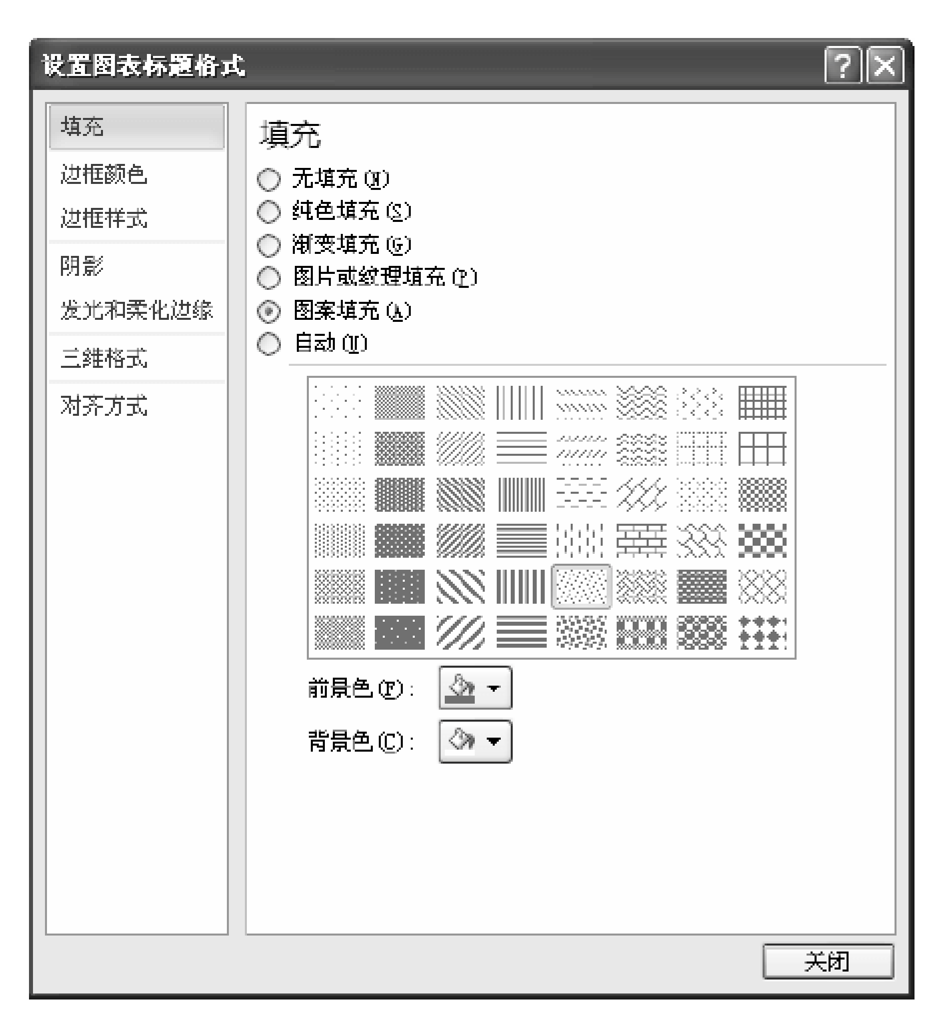


图 14-37　【设置图表标题格式】对话框

11 单击【关闭】按钮，即可为图表标题设置图案填充效果，如图 14-38 所示。

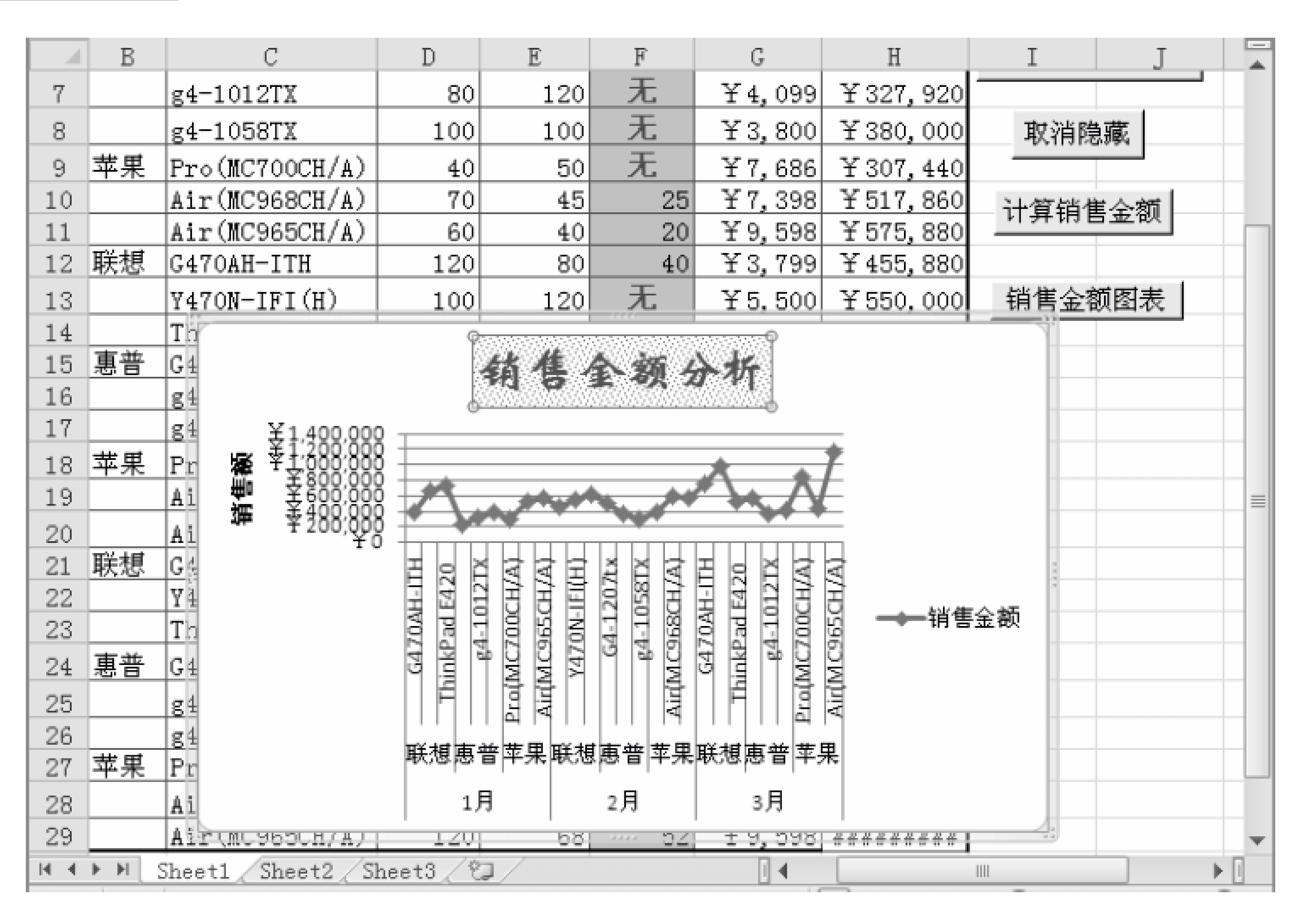


图 14-38　图案填充效果

到此就完成了对销售金额的图表分析。但是使用代码会涉及很多的格式参数，这些参数不方便记忆，所以完全使用程序代码添加图标会显得麻烦，而且还容易出现错误。这里，用户可以使用录制宏的方法，将插入图表的整个操作过程录制下来，然后将录制的代码复制到相应的子过程中，并且还可以根据实际需要对代码进行相应的调整或更改。这样既免去了向VBA代码窗口中输入代码的麻烦，又可以避免输入代码时造成的错误。

使用录制宏的方法编辑插入图表代码的具体操作步骤如下。

01 选择工作表中功能区的【视图】选项卡，选择【宏】选项组中的【宏】➤【录制宏】菜单命令，打开【录制新宏】对话框，单击【确定】按钮，开始录制宏，如图 14-39 所示。

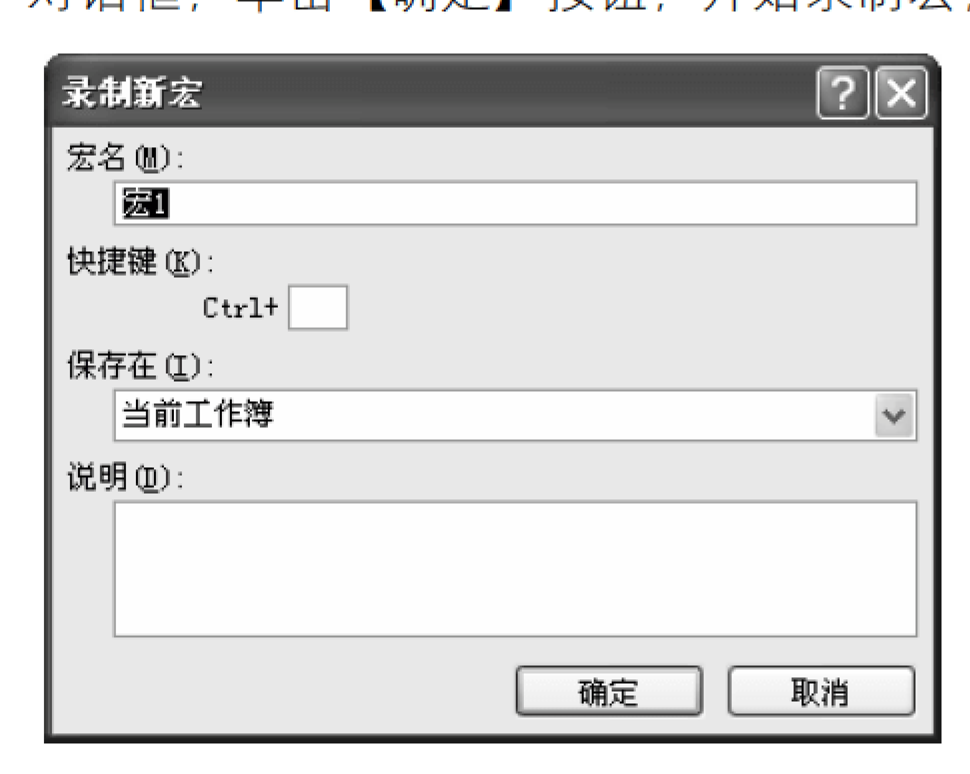


图 14-39　【录制新宏】对话框

02 按住 Ctrl 键的同时，依次单击需要用图表表示的单元格，作为待插入图表的数据源，这里选择的数据源为 A2:C29 和 H2:H29，如图 14-40 所示。



	A	B	C	D	E	F	G	H
1	2012年第一季度商品需求量及销售情况分析							
2	日期	品牌	产品型号	订购量	库存量	需求量	单价	销售金额
3	1月	联想	G470AH-ITH	100	200	无	￥3,799	￥379,900
4			Y470N-IFI(H)	120	100	20	￥5,500	￥660,000
5			ThinkPad E420	150	80	70	￥4,799	￥719,850
6		惠普	G4-1207tx	60	120	无	￥3,849	￥230,940
7			g4-1012TX	80	120	无	￥4,099	￥327,920
8			g4-1058TX	100	100	无	￥3,800	￥380,000
9		苹果	Pro(MC700CH/A)	40	50	无	￥7,686	￥307,440
10			Air(MC968CH/A)	70	45	25	￥7,398	￥517,860
11			Air(MC965CH/A)	60	40	20	￥9,598	￥575,880
12	2月	联想	G470AH-ITH	120	80	40	￥3,799	￥455,880
13			Y470N-IFI(H)	100	120	无	￥5,500	￥550,000
14			ThinkPad E420	130	100	30	￥4,799	￥623,870
15		惠普	G4-1207tx	130	90	40	￥3,849	￥500,370
16			g4-1012TX	90	50	40	￥4,099	￥368,910
17			g4-1058TX	80	55	25	￥3,800	￥304,000
18		苹果	Pro(MC700CH/A)	50	100	无	￥7,686	￥384,300
19			Air(MC968CH/A)	80	64	16	￥7,398	￥591,840
20			Air(MC965CH/A)	60	80	无	￥9,598	￥575,880
21	3月	联想	G470AH-ITH	200	165	35	￥3,799	￥759,800
22			Y470N-IFI(H)	180	126	54	￥5,500	￥990,000
23			ThinkPad E420	110	96	14	￥4,799	￥527,890

图 14-40　选择数据源

03 数据源选择完毕后，选择功能区【插入】选项卡，在【图表】下拉列表中选择【折线图】，显示出折线图类型，如图 14-41 所示。

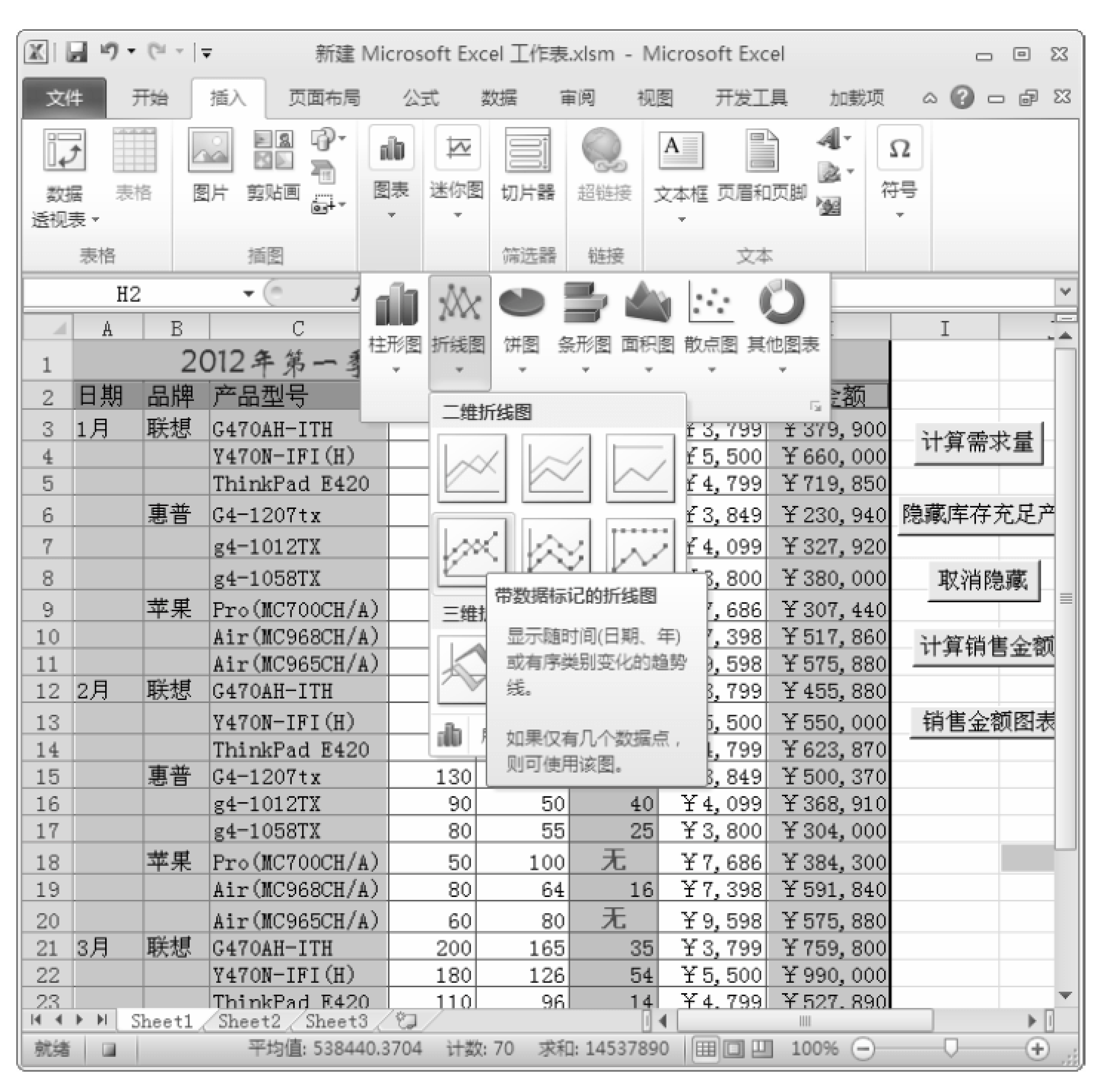


图 14-41　折线图类型

04 从中单击需要的图表类型，可在工作表中插入由所选数据源组成的图表，如图 14-42 所示。

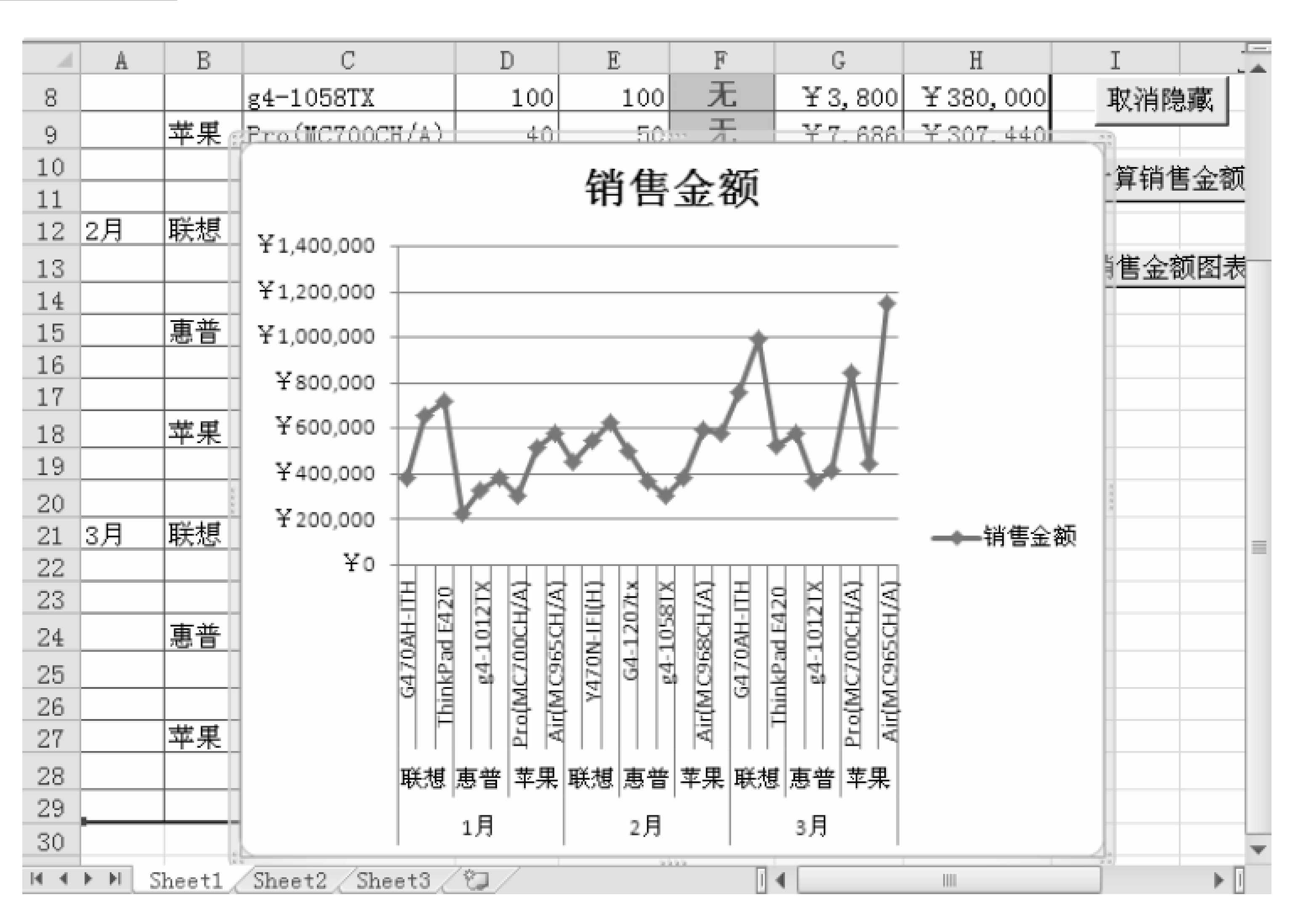


图 14-42　插入图表

05 选择功能区【视图】选项卡，在【宏】选项组中选择【宏】➤【停止录制】菜单命令，完成宏的录制操作。

06 在【宏】选项组中选择【宏】➤【查看宏】菜单命令，打开【宏】对话框，在【宏名】列表框中选择录制的宏——“宏 1”，如图 14-43 所示。

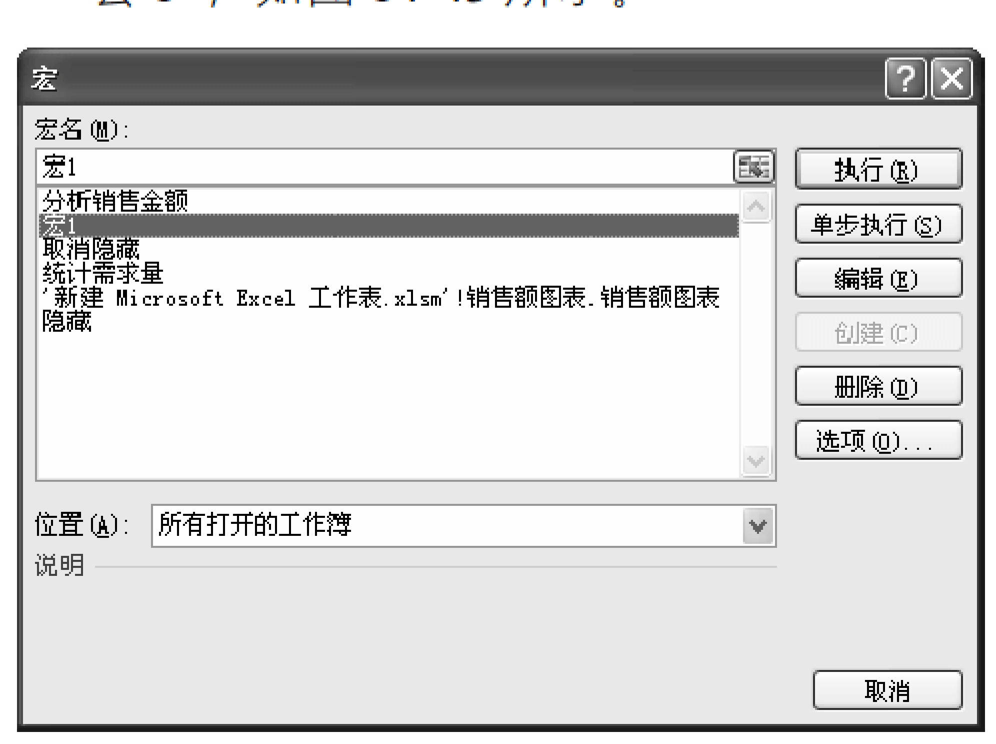


图 14-43　【宏】对话框

07 单击【编辑】按钮，可看到录制的宏代码，如图 14-44 所示。

```
(通用)                                   宏1
Sub 宏1()
'
' 宏1 宏
'

'
    ActiveWindow.SmallScroll Down:=-36
    Range("A2:C29").Select
    ActiveWindow.SmallScroll Down:=-42
    Range("A2:C29,H2:H29").Select
    Range("H2").Activate
    ActiveSheet.Shapes.AddChart.Select
    ActiveChart.ChartType = xlLineMarkers
    ActiveChart.SetSourceData Source:=Range("Sheet1!$A$2:$C$29,Sheet1!$H$2:$H$29"
        )
End Sub
```

图 14-44　录制的宏代码

通过以上录制宏的操作，不仅可以方便地得到插入图表的 VBA 程序代码，还避免了向 VBA 窗口中输入代码的麻烦以及输入错误的发生。



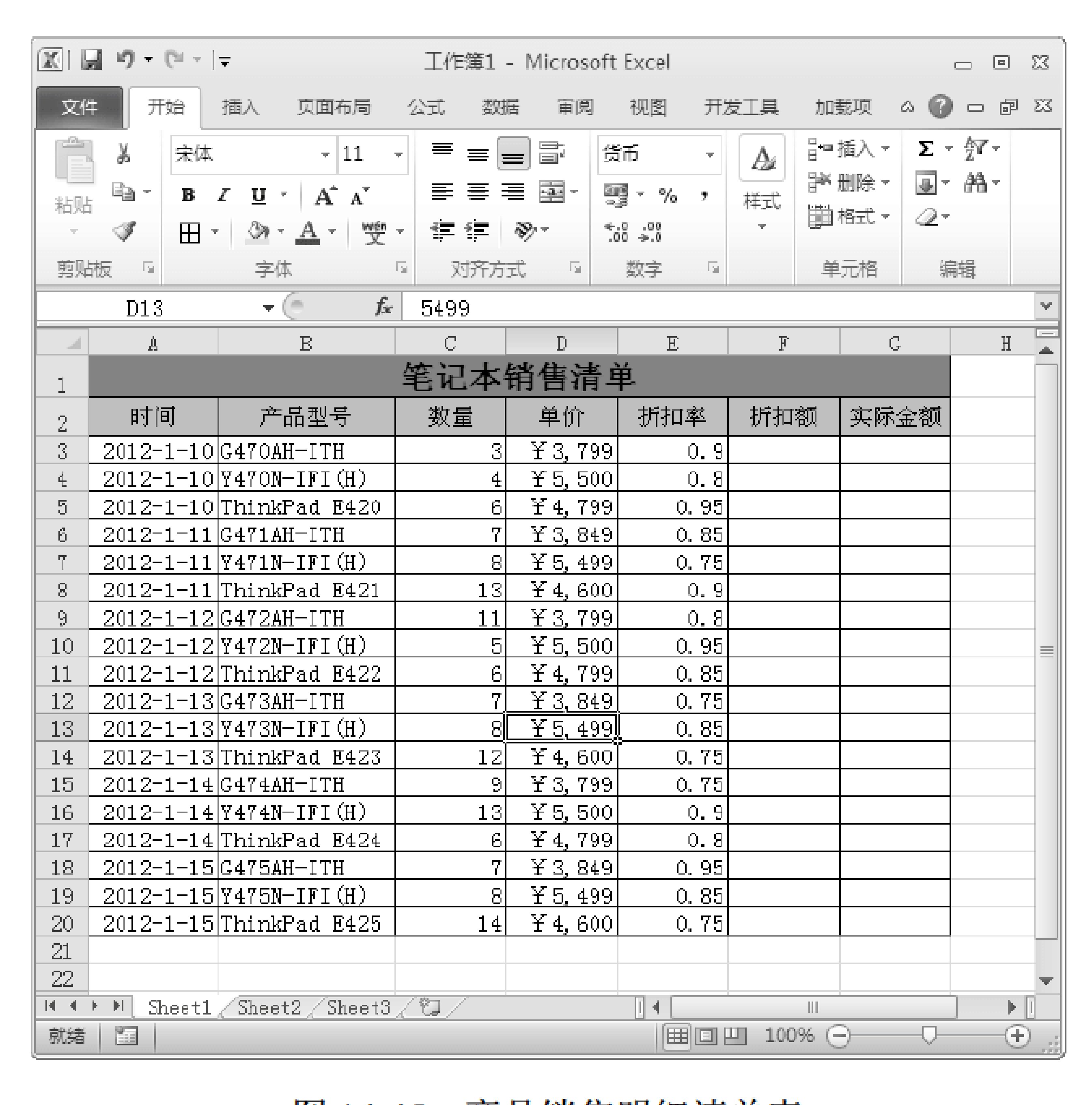
14.2 销售明细清单分析

销售明细清单主要记录了商品的销售记录信息，包括销售时间、产品名、数量、价格和折扣等内容。统计销售清单能让经营者更清晰地掌握当前的销售状况，对下一步销售策略的调整很有帮助。下面详细介绍统计与分析销售明细清单的方法。

14.2.1 统计销售明细清单

商品销售过程中会产生很多零碎的销售明细，根据销售的商品不同产生的销售清单也各有差异，但是大多数销售明细清单的数据信息都比较庞大，使用常规的 Excel 数据处理功能会比较麻烦，但是使用 Excel VBA 可以在很短的时间内完成对大量数据表格的格式化和数据查询等操作。

下面以如图 14-45 所示的笔记本销售清单为例，介绍使用 Excel VBA 格式化销售明细清单的方法。



笔记本销售清单

时间	产品型号	数量	单价	折扣率	折扣额	实际金额
2012-1-10	G470AH-ITH	3	￥3,799	0.9		
2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8		
2012-1-10	ThinkPad E420	6	￥4,799	0.95		
2012-1-11	G471AH-ITH	7	￥3,849	0.85		
2012-1-11	Y471N-IFI(H)	8	￥5,499	0.75		
2012-1-11	ThinkPad E421	13	￥4,600	0.9		
2012-1-12	G472AH-ITH	11	￥3,799	0.8		
2012-1-12	Y472N-IFI(H)	5	￥5,500	0.95		
2012-1-12	ThinkPad E422	6	￥4,799	0.85		
2012-1-13	G473AH-ITH	7	￥3,849	0.75		
2012-1-13	Y473N-IFI(H)	8	￥5,499	0.85		
2012-1-13	ThinkPad E423	12	￥4,600	0.75		
2012-1-14	G474AH-ITH	9	￥3,799	0.75		
2012-1-14	Y474N-IFI(H)	13	￥5,500	0.9		
2012-1-14	ThinkPad E424	6	￥4,799	0.8		
2012-1-15	G475AH-ITH	7	￥3,849	0.95		
2012-1-15	Y475N-IFI(H)	8	￥5,499	0.85		
2012-1-15	ThinkPad E425	14	￥4,600	0.75		

图 14-45　商品销售明细清单表

具体的操作步骤如下。



01 打开“笔记本销售清单”工作表 Sheet1，选择功能区【开发工具】选项卡，单击【代码】选项组中的【Visual Basic】按钮，打开【Microsoft Visual Basic for Application】主窗口，在菜单栏选择【插入】➤【模块】菜单命令，插入一个模块，在该模块的【名称】属性列表中将其更改为“统计金额”，如图 14-46 所示。

02 双击进入该模块代码窗口中，在菜单栏选择【插入】➤【过程】菜单命令，打开【添加过程】对话框，在【名称】文本框中输入“统计金额”，在【类型】选项组中选中【子过程】单选按钮，在【范围】选项组中选中【公共的】单选按钮，如图 14-47 所示。



图 14-46 插入模块并更改其名称

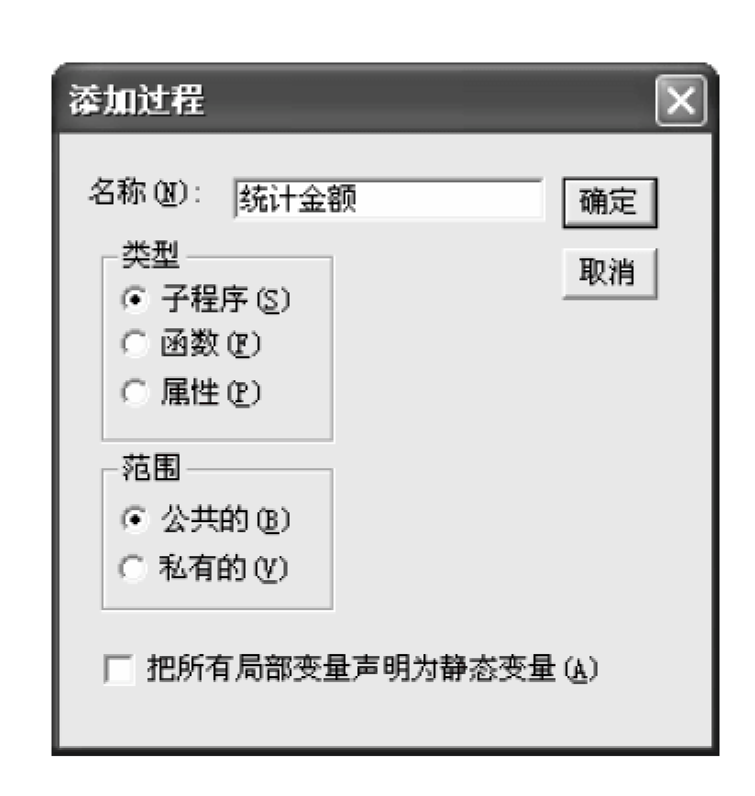


图 14-47 【添加过程】对话框

03 单击【确定】按钮，在模块中创建名为“统计金额”公有子过程的代码框架，如图 14-48 所示。

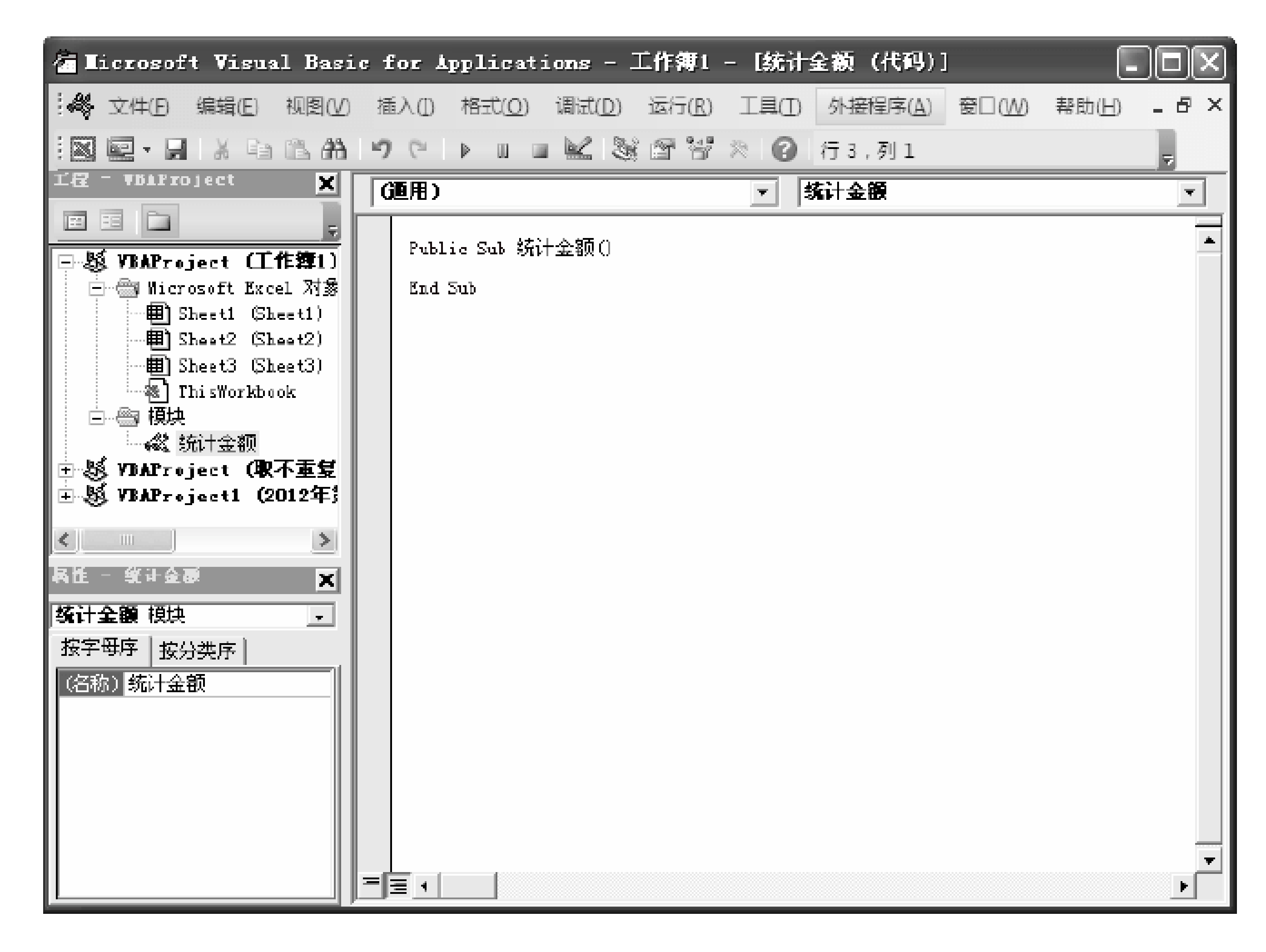


图 14-48 创建“统计金额”子过程代码框架

04 在该子过程中输入以下程序代码，如图 14-49 所示。

```
Public Sub 统计金额()
Dim i As Integer                    '定义用于循环的整型变量
```

```
    Dim count As Integer              '定义用于记录工作表的记录数的整型变量
    count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count
                                    '把工作表的记录数赋给整型变量 count
      For i = 3 To count
         Cells(i, 6) = Cells(i, 4) * Cells(i, 3) * (1 - Cells(i, 5))
                                  '折扣额=单价*数量*（1-折扣率）
         Cells(i, 7) = Cells(i, 4) * Cells(i, 3) * Cells(i, 5)
                                   '实际金额=单价*数量*折扣率
      Next i
    End Sub
```

```
(通用)                                  统计金额

Public Sub 统计金额()
Dim i As Integer                        '定义用于循环的整型变量
Dim count As Integer                 '定义用于记录工作表的记录数的整型变量
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count
                                    '把工作表的记录数赋给整型变量count
  For i = 3 To count
     Cells(i, 6) = Cells(i, 4) * Cells(i, 3) * (1 - Cells(i, 5))
                               '折扣额=单价*数量*(1-折扣率)
     Cells(i, 7) = Cells(i, 4) * Cells(i, 3) * Cells(i, 5)
                                '实际金额=单价*数量*折扣率
  Next i
End Sub
```

图 14-49　输入统计金额的程序代码

【提示】这段代码首先使用整型变量 count 记录工作表 Sheet1 记录数，然后在 For 循环语句中分别计算"折扣额"和"实际金额"。

05 代码输入完毕后，在菜单栏选择【运行】➢【运行子过程/用户窗体】菜单命令，运行该子过程，计算出折扣额及实际金额的结果，如图 14-50 所示。

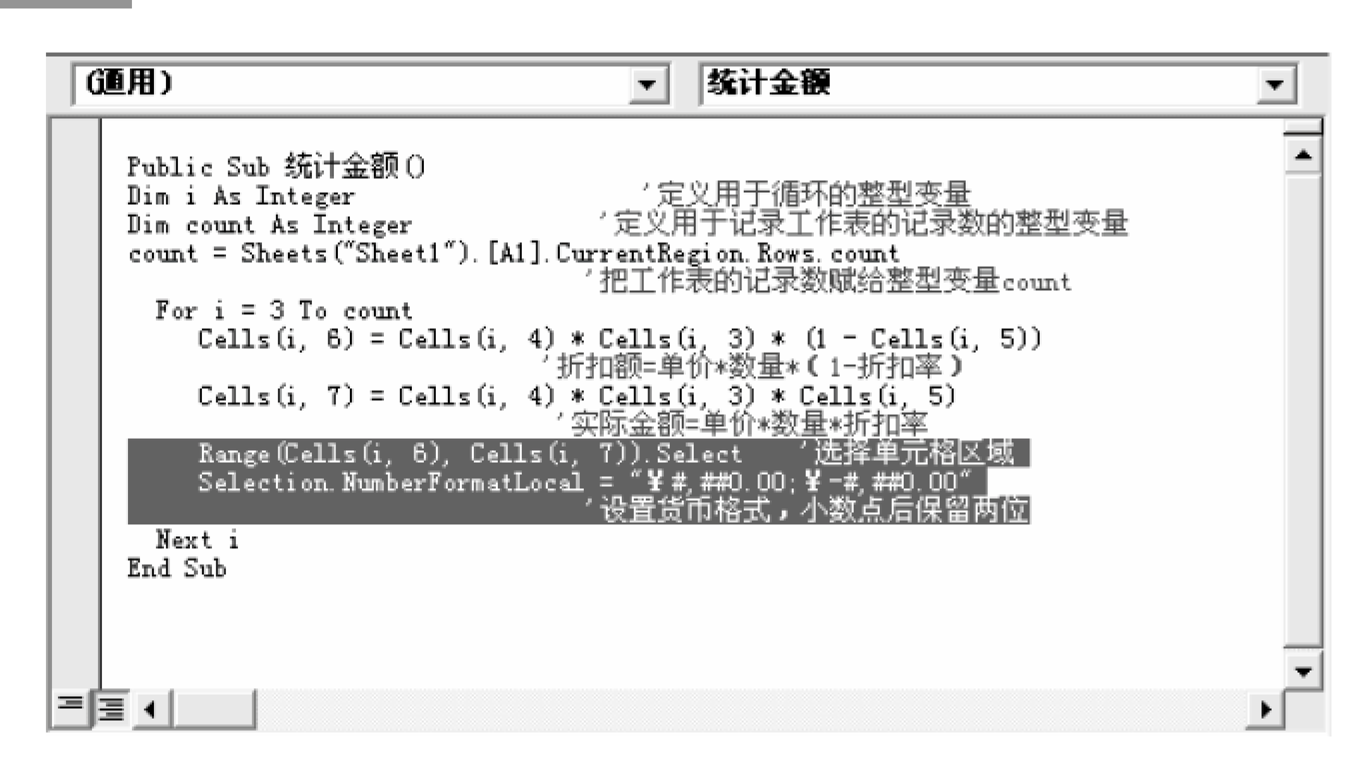
	A	B	C	D	E	F	G	H
1	笔记本销售清单							
2	时间	产品型号	数量	单价	折扣率	折扣额	实际金额	
3	2012-1-10	G470AH-ITH	3	￥3,799	0.9	1139.7	10257.3	
4	2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8	4400	17600	
5	2012-1-10	ThinkPad E420	6	￥4,799	0.95	1439.7	27354.3	
6	2012-1-11	G471AH-ITH	7	￥3,849	0.85	4041.45	22901.55	
7	2012-1-11	Y471N-IFI(H)	8	￥5,499	0.75	10998	32994	
8	2012-1-11	ThinkPad E421	13	￥4,600	0.9	5980	53820	
9	2012-1-12	G472AH-ITH	11	￥3,799	0.8	8357.8	33431.2	
10	2012-1-12	Y472N-IFI(H)	5	￥5,500	0.95	1375	26125	
11	2012-1-12	ThinkPad E422	6	￥4,799	0.85	4319.1	24474.9	
12	2012-1-13	G473AH-ITH	7	￥3,849	0.75	6735.75	20207.25	
13	2012-1-13	Y473N-IFI(H)	8	￥5,499	0.85	6598.8	37393.2	
14	2012-1-13	ThinkPad E423	12	￥4,600	0.75	13800	41400	
15	2012-1-14	G474AH-ITH	9	￥3,799	0.75	8547.75	25643.25	
16	2012-1-14	Y474N-IFI(H)	13	￥5,500	0.9	7150	64350	
17	2012-1-14	ThinkPad E424	6	￥4,799	0.8	5758.8	23035.2	
18	2012-1-15	G475AH-ITH	7	￥3,849	0.95	1347.15	25595.85	
19	2012-1-15	Y475N-IFI(H)	8	￥5,499	0.85	6598.8	37393.2	
20	2012-1-15	ThinkPad E425	14	￥4,600	0.75	16100	48300	
21								

图 14-50　计算结果

06 在图 14-50 中可以看到，计算出的"折扣额"和"实际金额"的输出结果都不是货币格式，所以还需要将其转化为货币格式，则需要在上述过程中添加以下代码，如图 14-51 所示。

```
    Range(Cells(i, 6), Cells(i, 7)).Select   '选择单元格区域
    Selection.NumberFormatLocal = "￥#,##0.00;￥-#,##0.00"   '设置货币格式，小数点后
'保留两位
```





```
(通用)                                   统计金额

Public Sub 统计金额()
Dim i As Integer                    '定义用于循环的整型变量
Dim count As Integer             '定义用于记录工作表的记录数的整型变量
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count
                                 '把工作表的记录数赋给整型变量count
  For i = 3 To count
     Cells(i, 6) = Cells(i, 4) * Cells(i, 3) * (1 - Cells(i, 5))
                              '折扣额=单价*数量*(1-折扣率)
     Cells(i, 7) = Cells(i, 4) * Cells(i, 3) * Cells(i, 5)
                               '实际金额=单价*数量*折扣率
     Range(Cells(i, 6), Cells(i, 7)).Select     '选择单元格区域
     Selection.NumberFormatLocal = "￥#,##0.00;￥-#,##0.00"
                                 '设置货币格式，小数点后保留两位
  Next i
End Sub
```

图 14-51　添加转化货币格式的代码

06 添加完毕后，在菜单栏选择【运行】➤【运行子过程/用户窗体】菜单命令，运行该子过程，将计算结果转化为货币格式，如图 14-52 所示。

	A	B	C	D	E	F	G
1	笔记本销售清单						
2	时间	产品型号	数量	单价	折扣率	折扣额	实际金额
3	2012-1-10	G470AH-ITH	3	￥3,799	0.9	￥1,139.70	￥10,257.30
4	2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00
5	2012-1-10	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30
6	2012-1-11	G471AH-ITH	7	￥3,849	0.85	￥4,041.45	￥22,901.55
7	2012-1-11	Y471N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00
8	2012-1-11	ThinkPad E421	13	￥4,600	0.9	￥5,980.00	￥53,820.00
9	2012-1-12	G472AH-ITH	11	￥3,799	0.8	￥8,357.80	￥33,431.20
10	2012-1-12	Y472N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00
11	2012-1-12	ThinkPad E422	6	￥4,799	0.85	￥4,319.10	￥24,474.90
12	2012-1-13	G473AH-ITH	7	￥3,849	0.75	￥6,735.75	￥20,207.25
13	2012-1-13	Y473N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20
14	2012-1-13	ThinkPad E423	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	2012-1-14	G474AH-ITH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	2012-1-14	Y474N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	2012-1-14	ThinkPad E424	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	2012-1-15	G475AH-ITH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	2012-1-15	Y475N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20
20	2012-1-15	ThinkPad E425	14	￥4,600	0.75	￥16,100.00	￥48,300.00

图 14-52　将计算结果转化为货币格式

14.2.2 分析销售明细清单

获取销售明细清单后，需要对销售明细清单进行分析处理，获取对商品销售经营有价值的信息。对商品销售情况进行分析时，可以通过设置“自动筛选”功能来实现，具体操作方法如下。

1. 统计 G470AH-ITH 笔记本的销售总额

统计 G470AH-ITH 笔记本的销售总额的具体操作步骤如下。

01 打开“笔记本销售清单”工作表，选择功能区【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中选择【命令按钮】按钮 （ActiveX 控件），然后在工作表的适当位置处，当鼠标指针变为“+”状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，插入一个命令按钮，如图 14-53 所示。



	B	C	D	E	F	G
1	笔记本销售清单					
2	产品型号	数量	单价	折扣率	折扣额	实际金额
3	G470AH-ITH	3	￥3,799	0.9	￥1,139.70	￥10,257.30
4	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00
5	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30
6	G471AH-ITH	7	￥3,849	0.85	￥4,041.45	￥22,901.55
7	Y471N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00
8	ThinkPad E421	13	￥4,600	0.9	￥5,980.00	￥53,820.00
9	G472AH-ITH	11	￥3,799	0.8	￥8,357.80	￥33,431.20
10	Y472N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00
11	ThinkPad E422	6	￥4,799	0.85	￥4,319.10	￥24,474.90
12	G473AH-ITH	7	￥3,849	0.75	￥6,735.75	￥20,207.25
13	Y473N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20
14	ThinkPad E423	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	G474AH-ITH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	Y474N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	ThinkPad E424	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	G475AH-ITH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	Y475N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20
20	ThinkPad E425	14	￥4,600	0.75	￥16,100.00	￥48,300.00

图 14-53　插入命令按钮

02 右击该按钮，在弹出的快捷菜单中选择【属性】菜单命令，打开【属性】窗口，在【名称】和 Caption 文本框中均输入"自动筛选"，如图 14-54 所示。

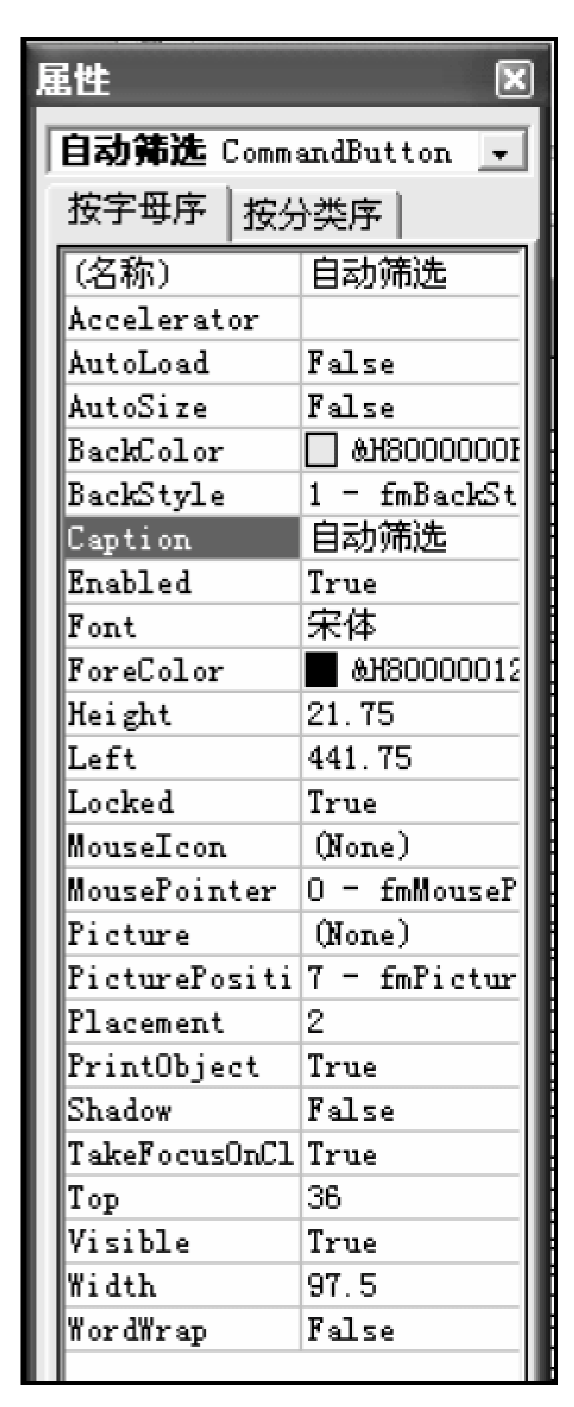


图 14-54　更改命令按钮属性值

03 属性设置完毕后，双击该按钮，打开 VBA 代码窗口，在其中输入以下实现自动筛选功能的程序代码，如图 14-55 所示。

```
Private Sub 自动筛选_Click()
Range("A2:G2").Select                '选择设置自动筛选的数据区域
Selection.AutoFilter                 '实现自动筛选功能
End Sub
```



```
Private Sub 自动筛选_Click()
Range("A2:G2").Select  '选择设置自动筛选的数据区域
Selection.AutoFilter   '实现自动筛选功能
End Sub
```

图 14-55　输入实现自动筛选功能的代码

04 代码输入完毕后，关闭代码窗口返回工作表中，单击功能区【开发工具】选项卡中的【设计模式】按钮，退出该按钮的编辑状态，此时单击【自动筛选】按钮可运行上述代码，实现自动筛选功能，如图 14-56 所示。

	A	B	C	D	E	F	G	H
1	笔记本销售清单							
2	时间	产品型号	数	单价	折扣	折扣额	实际金额	
3	2012-1-10	G470AH-ITH	3	￥3,799	0.9	￥1,139.70	￥10,257.30	自动筛选
4	2012-1-10	Y470N-IFI(H)	4	￥5,500	0.8	￥4,400.00	￥17,600.00	
5	2012-1-10	ThinkPad E420	6	￥4,799	0.95	￥1,439.70	￥27,354.30	
6	2012-1-11	G471AH-ITH	7	￥3,849	0.85	￥4,041.45	￥22,901.55	
7	2012-1-11	Y471N-IFI(H)	8	￥5,499	0.75	￥10,998.00	￥32,994.00	
8	2012-1-11	ThinkPad E421	13	￥4,600	0.9	￥5,980.00	￥53,820.00	
9	2012-1-12	G472AH-ITH	11	￥3,799	0.8	￥8,357.80	￥33,431.20	
10	2012-1-12	Y472N-IFI(H)	5	￥5,500	0.95	￥1,375.00	￥26,125.00	
11	2012-1-12	ThinkPad E422	6	￥4,799	0.85	￥4,319.10	￥24,474.90	
12	2012-1-13	G473AH-ITH	7	￥3,849	0.75	￥6,735.75	￥20,207.25	
13	2012-1-13	Y473N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
14	2012-1-13	ThinkPad E423	12	￥4,600	0.75	￥13,800.00	￥41,400.00	
15	2012-1-14	G474AH-ITH	9	￥3,799	0.75	￥8,547.75	￥25,643.25	
16	2012-1-14	Y474N-IFI(H)	13	￥5,500	0.9	￥7,150.00	￥64,350.00	
17	2012-1-14	ThinkPad E424	6	￥4,799	0.8	￥5,758.80	￥23,035.20	
18	2012-1-15	G475AH-ITH	7	￥3,849	0.95	￥1,347.15	￥25,595.85	
19	2012-1-15	Y475N-IFI(H)	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
20	2012-1-15	ThinkPad E425	14	￥4,600	0.75	￥16,100.00	￥48,300.00	
21								
22								
23								
24								

Sheet1　Sheet2　Sheet3

图 14-56　自动筛选结果

05 在工作表的单元格 H2 中输入要统计的字段名称，如 "G470AH 销售总额"，并在上述过程代码后面添加下面的程序代码，如图 14-57 所示。

```
Public Sub  G470AH 销售总额()
Dim i As Integer '定义用于循环的整型变量
Dim count As Integer '定义用于保存工作表记录数的整型变量
Dim temp As Variant  '定义用于存放总金额的变体型变量
Sheets("Sheet1").Select
Selection.AutoFilter Field:=2, Criteria1:="G470AH"
                 '在第 2 列以“G470AH”为条件进行筛选
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count
                      '把工作表的记录数赋给 count
temp = 0  '为其赋初值
For i = 3 To count
    Rows(i).Select  '选择第 i 行
    If Selection.EntireRow.Hidden = False Then
```

```
                '判断第 i 行是否隐藏，如果不隐藏则进行下列计算
        temp = temp + Cells(i, 7) '计算总和
    End If
Next i
Cells(3 8) = temp  '将累加的总金额赋值给单元格 H3
Cells(3, 8).Select  '选择单元格 H3
     With Selection
         .NumberFormatLocal = "￥#,##0.00; ￥-#,##0.00"  '设置为【货币】格式
         .HorizontalAlignment = xlCenter '设置居中对齐
         .VerticalAlignment = xlCenter
     End With
End Sub
```

```
(通用)    G470AH销售总额

Private Sub 自动筛选_Click()
Range("A2:G2").Select  '选择设置自动筛选的数据区域
Selection.AutoFilter   '实现自动筛选功能
End Sub

Public Sub G470AH销售总额()
Dim i As Integer '定义用于循环的整型变量
Dim count As Integer '定义用于保存工作表记录数的整型变量
Dim temp As Variant  '定义用于存放总金额的变体型变量
Sheets("Sheet1").Select
Selection.AutoFilter Field:=2, Criteria1:="G470AH"
                    '在第2列以"G470AH"为条件进行筛选
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count
                          '把工作表的记录数赋给count
temp = 0  '为其赋初值
For i = 3 To count
    Rows(i).Select  '选择第i行
    If Selection.EntireRow.Hidden = False Then
                 '判断第i行是否隐藏，如果不隐藏则进行下列计算
        temp = temp + Cells(i, 7) '计算总和
    End If
Next i
Cells(4, 8) = temp  '将累加的总金额赋值给单元格H4
Cells(4, 8).Select  '选择单元格H4
     With Selection
         .NumberFormatLocal = "￥#,##0.00;￥-#,##0.00"  '设置为【货币】格式
         .HorizontalAlignment = xlCenter '设置居中对齐
         .VerticalAlignment = xlCenter
     End With
End Sub
```

图 14-57　输入计算 G470AH 的销售总额代码

06 代码添加完毕后，在菜单栏选择【运行】➢【运行子过程/用户窗体】菜单命令，运行该子过程，计算出 G470AH 笔记本的销售总额，并将结果显示在 H3 单元格中，如图 14-58 所示。

	A	B	C	D	E	F	G	H
1	笔记本销售清单							
2	时间	产品型	数	单价	折扣	折扣额	实际金额	G470AH销售总
3	2012-1-10	G470AH	3	￥3,799	0.9	￥1,139.70	￥10,257.30	￥138,036.40
6	2012-1-11	G470AH	7	￥3,849	0.85	￥4,041.45	￥22,901.55	自动筛选
9	2012-1-12	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20	
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25	
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25	
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85	
21								
22								
23								
24								
25								
26								
27								
28								
29								

Sheet1　Sheet2　Sheet3

图 14-58　计算出 G470AH 的销售总额

【提示】在统计 G470AH 笔记本实际销售总额时，会将其他产品行隐藏，如果 H3 行不是 G470AH 笔记本，则该行隐藏，获得的销售总额也将无法看到，所以输出结果的单元格最好在要统计的对象所在的行。

用户也可以按照同样的方法来统计其他字段的销售总额或总销售数量，只需在语句“Selection.AutoFilter Field:=2, Criteria1:=" G470AH"”中转换相应的列号及其筛选条件即可。

2. 统计折扣额不低于 5000 的记录

在商品销售清单数据表中，自定义筛选“折扣额”大于 5000 的商品记录，具体操作步骤如下。

01 单击如图 14-59 所示的“折扣额”字段所在单元格右侧的下三角按钮，在弹出的下拉菜单中选择【数字筛选】➤【自定义筛选】菜单命令。

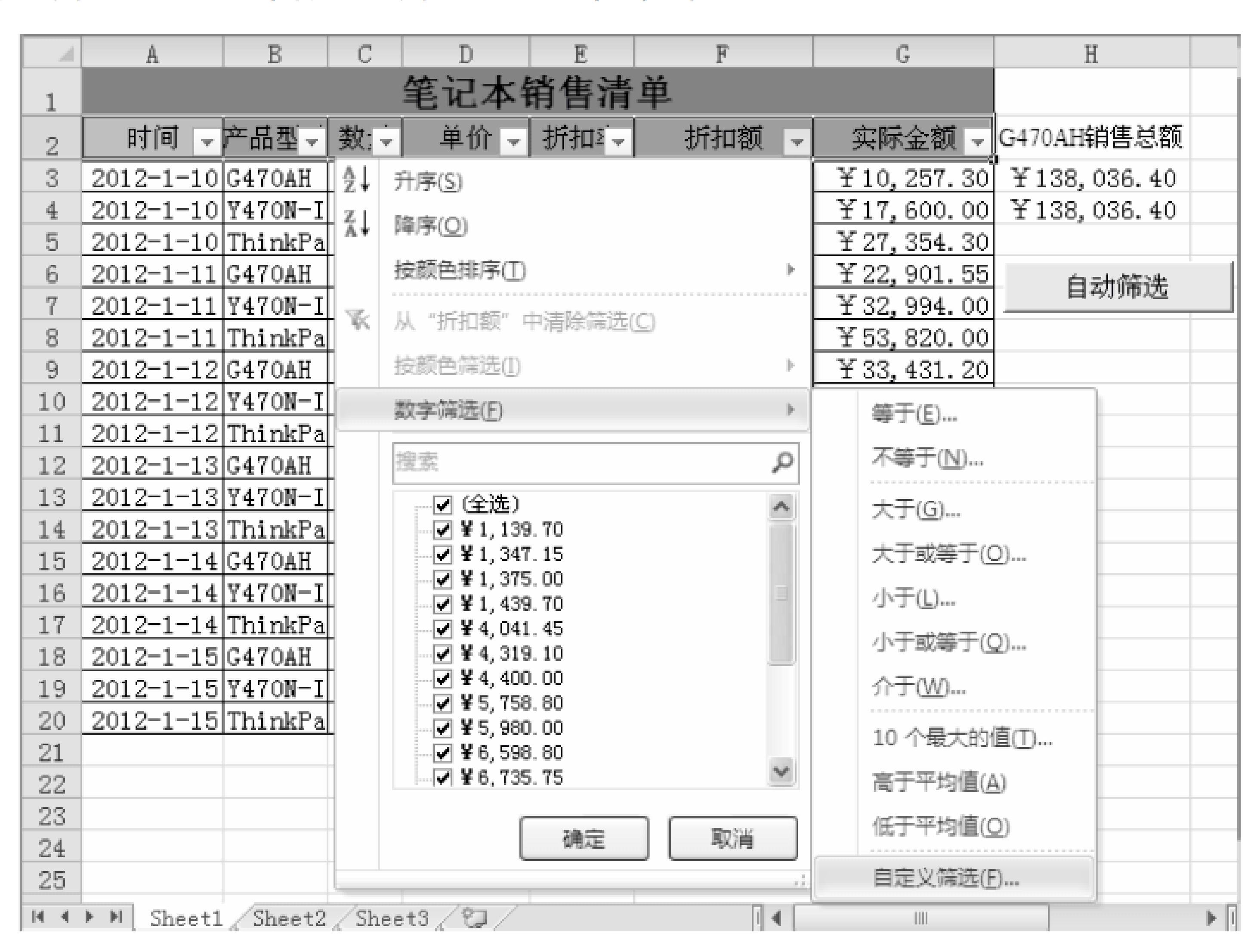


图 14-59　选择自定义选项

02 打开【自定义自动筛选方式】对话框，在【折扣额】组合框中的第 1 个下拉列表中选择“大于或等于”，在第 2 个下拉列表中输入“5000”，如图 14-60 所示。

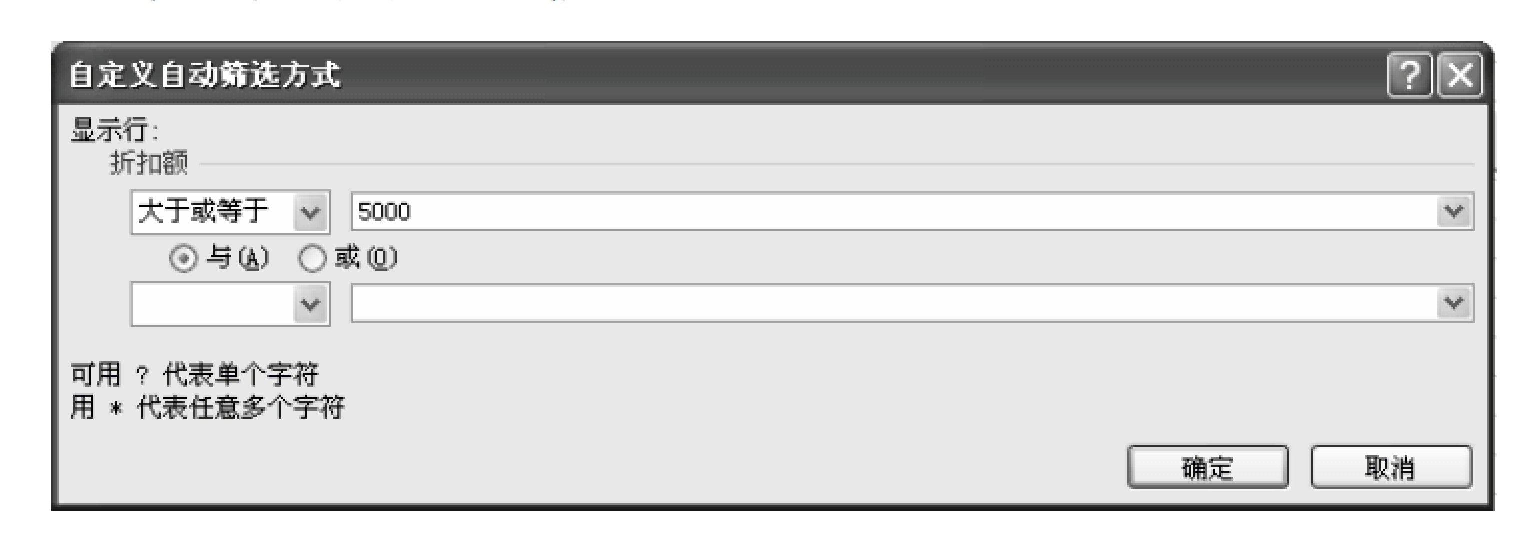


图 14-60　【自定义自动筛选方式】对话框

03 单击【确定】按钮，可看到自定义筛选“折扣额”的结果，如图 14-61 所示。

	A	B	C	D	E	F	G	H
1	笔记本销售清单							
2	时间	产品型	数:	单价	折扣:	折扣额	实际金额	G470AH销售总额
7	2012-1-11	Y470N-I	8	￥5,499	0.75	￥10,998.00	￥32,994.00	自动筛选
8	2012-1-11	ThinkPa	13	￥4,600	0.9	￥5,980.00	￥53,820.00	
9	2012-1-12	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20	
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25	
13	2012-1-13	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
14	2012-1-13	ThinkPa	12	￥4,600	0.75	￥13,800.00	￥41,400.00	
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25	
16	2012-1-14	Y470N-I	13	￥5,500	0.9	￥7,150.00	￥64,350.00	
17	2012-1-14	ThinkPa	6	￥4,799	0.8	￥5,758.80	￥23,035.20	
19	2012-1-15	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
20	2012-1-15	ThinkPa	14	￥4,600	0.75	￥16,100.00	￥48,300.00	
21								
22								

图 14-61　自定义筛选结果

在实际应用中，如果有很多数据表需要进行此类自定义自动筛选操作，则可以使用 Excel VBA 来提高工作效率和准确度。

使用 Excel VBA 实现上述自定义自动筛选功能的方法是，在模块中插入一个公有子过程，然后在该子过程中添加实现自动筛选功能的语句，具体程序代码如下，如图 14-62 所示。

```
    Public Sub 条件筛选()
    Selection.AutoFilter   field:=7,   Criteria1:=">=35000",   Operator:=xlAnd,
Criteria2:="<=50000"
           '在第 7 列以“大于或等于 35000 且小于或等于 50000”为条件进行筛选
    End Sub
```



图 14-62　输入自动筛选功能代码

在这段代码中，field:=7 用于设置对第 7 列（“实际金额”所在列）进行自动筛选操作；Criteria1:= “>=35000” 用于设置第 1 个筛选条件是 “>=35000”；Operator:=xlAnd 用于设置多个筛选条件之间是“逻辑与”关系；Criteria2:= “<=50000” 用于设置第 2 个筛选条件是 “<=50000”。

3. 统计“销售数量”不低于 8 且“折扣率”不低于 0.85 的商品记录

经常会遇到特定信息的筛选，例如要统计“销售数量”不低于 8 且“折扣率”不低于 0.85 的商品记录。实现该统计需求的方法主要有两种，一种是利用 Excel VBA 代码实现，另一种

则是利用工作表中的菜单命令实现。

1）利用 VBA 实现

利用 Excel VBA 代码实现高级筛选功能的具体操作步骤如下。

01 在“笔记本销售清单”工作表中空白单元格建立用于存放高级筛选条件的单元格区域，如图 14-63 所示。

	A	B	C	D	E	F	G
7	2012-1-11	Y470N-I	8	￥5,499	0.75	￥10,998.00	￥32,994.00
8	2012-1-11	ThinkPa	13	￥4,600	0.9	￥5,980.00	￥53,820.00
9	2012-1-12	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20
10	2012-1-12	Y470N-I	5	￥5,500	0.95	￥1,375.00	￥26,125.00
11	2012-1-12	ThinkPa	6	￥4,799	0.85	￥4,319.10	￥24,474.90
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25
13	2012-1-13	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20
14	2012-1-13	ThinkPa	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	2012-1-14	Y470N-I	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	2012-1-14	ThinkPa	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	2012-1-15	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20
20	2012-1-15	ThinkPa	14	￥4,600	0.75	￥16,100.00	￥48,300.00
21							
22				销售数量	折扣率		
23				>=8	>=0.85		

图 14-63　建立条件区域效果

02 选择工作表功能区的【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮，打开 Microsoft Visual Basic for Application 主窗口，在菜单栏选择【插入】➢【模块】菜单命令，插入一个模块，在该模块的【名称】属性列表中将其名称更改为“高级筛选”。然后双击该模块，在菜单栏选择【插入】➢【子过程】菜单命令，打开【添加过程】对话框，在【名称】文本框中输入“高级筛选”，并选中【子程序】和【公共的】单选按钮，如图 14-64 所示。



图 14-64　【添加过程】对话框

03 单击【确定】按钮，创建名为“高级筛选”的公有子过程，并输入以下程序代码，如图 14-65 所示。

```
Public Sub 高级筛选()
Range("A2:G20").AdvancedFilter _      '设置高级筛选的数据源区域为A2:G20
Action:=xlFilterInPlace, CriteriaRange:=Range("C23:D24"), Unique:=False
```

```
                                '设置高级筛选的条件区域为 C23:D24
End Sub
```



图 14-65　输入高级筛选代码程序

【提示】其中 Action:=xlFilterInPlace 用于将参数 Action 的类型值设置为 xlFilterInPlace。Unique:=False 用于设置筛选结果的唯一性，如果赋值为 True，则将重复出现的记录仅保留一条；如果赋值为 False，则将筛选出所有符合条件的记录。该参数的默认值为 False。

04 关闭代码窗口，返回工作表，为该过程设置执行按钮。选择功能区【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中选择【按钮】按钮（窗体控件），然后在工作表的适当位置，当鼠标指针变为“+”状时，按住鼠标左键不放并拖动至适当大小后释放鼠标左键，系统则弹出【指定宏】对话框，在【宏名】列表中选择“高级筛选”选项，如图 14-66 所示。

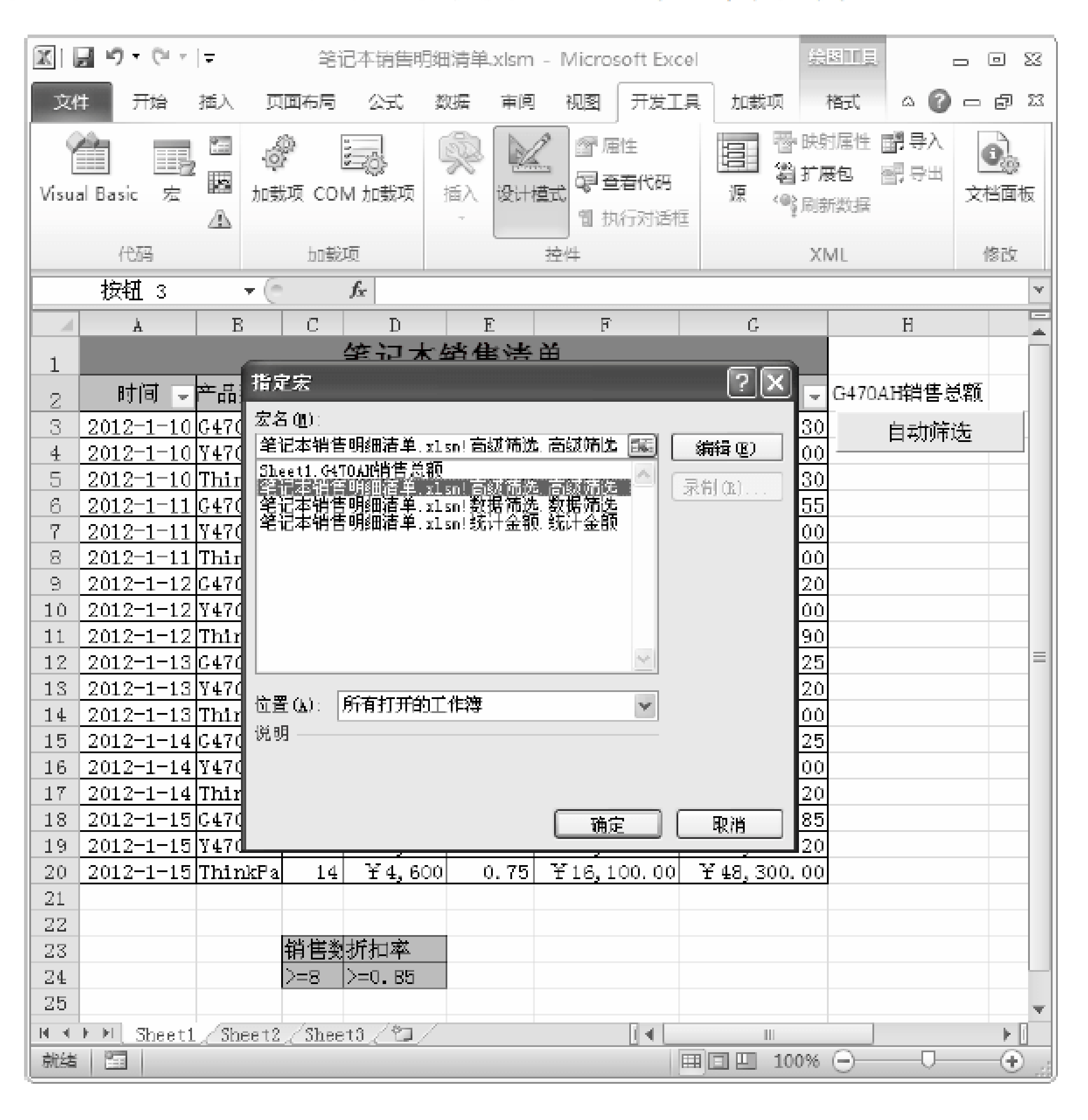


图 14-66　【指定宏】对话框



05 单击【确定】按钮，完成宏的指定操作。右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为"高级筛选"，然后单击工作表中的任意单元格，退出该按钮的编辑状态，如图 14-67 所示。



图 14-67　更改按钮名称

06 单击【高级筛选】按钮，执行指定子过程，进行高级筛选操作，如图 14-68 所示。

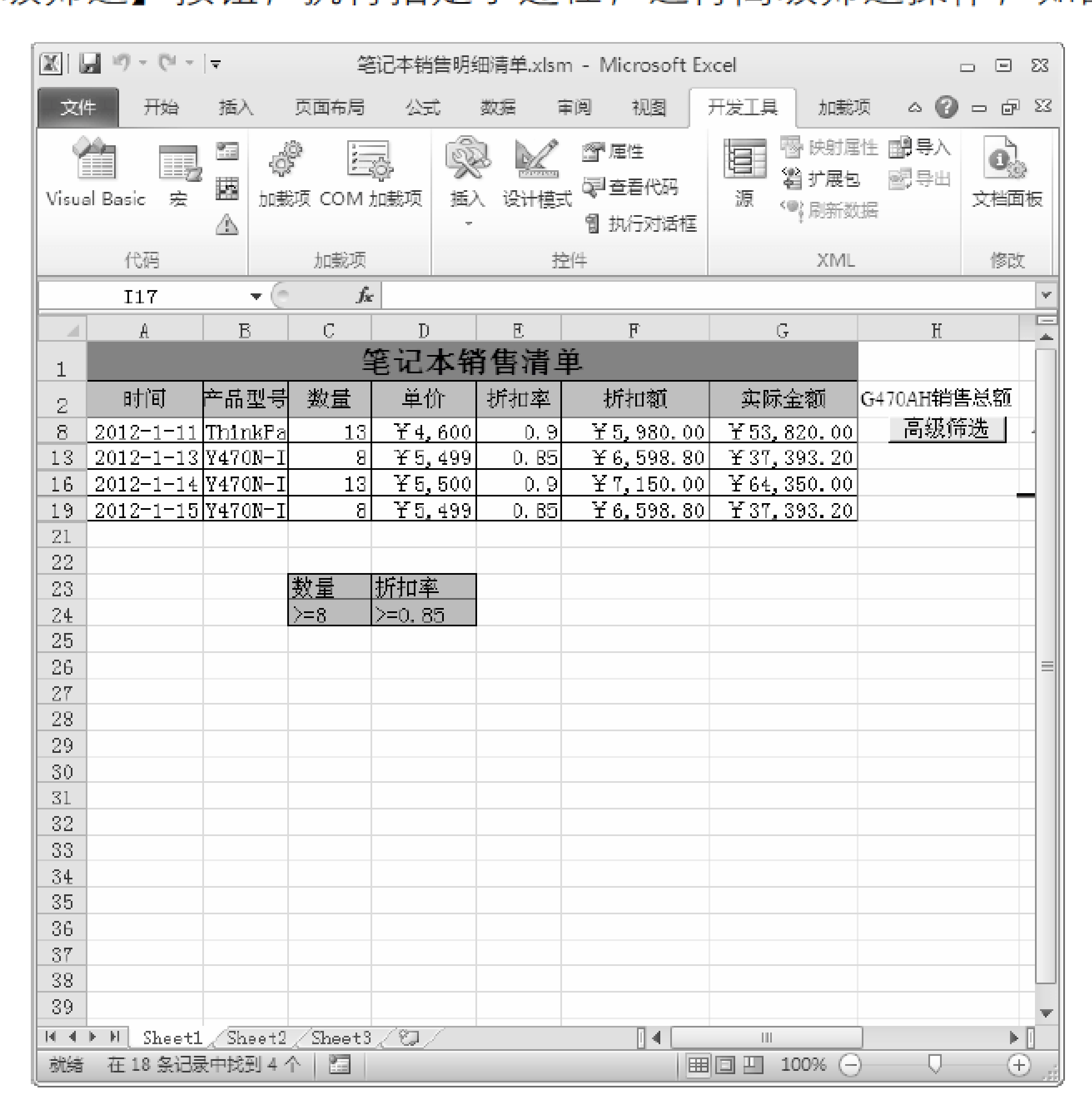


图 14-68　高级筛选结果

2）利用菜单命令实现高级筛选功能

利用菜单命令实现高级筛选功能的具体操作步骤如下。

01 在设置完用于存放高级筛选条件的单元格区域后，选择功能区【数据】选项卡，单击【排序和筛选】选项组中的【高级】按钮，打开【高级筛选】对话框，如图 14-69 所示。



图 14-69　【高级筛选】对话框

02 在该对话框中的【方式】选项区中选择【在原有区域显示筛选结果】单选按钮，单击【列表区域】文本框右侧的折叠按钮，然后在工作表中选择 A2:G20 单元格区域，如图 14-70 所示。

	A	B	C	D	E	F	G	H
1	笔记本销售清单							
2	时间	产品型	数量	单价	折扣3	折扣额	实际金额	G470AH销售总额
3	2012-1-10	G470AH	3	￥3,799	0.9	￥1,139.70	￥10,257.30	￥138,036.40
4	2012-1-10	Y470N-I	4	￥5,500	0.8	￥4,400.00	￥17,600.00	
5	2012-1-10	ThinkPa	6	￥4,799	0.			自动筛选
6	2012-1-11	G470AH	7	￥3,849	0.			
7	2012-1-11	Y470N-I	8	￥5,499	0.			
8	2012-1-11	ThinkPa	13	￥4,600	0.9	￥5,980.00	￥53,820.00	
9	2012-1-12	G470AH	11	￥3,799	0.8	￥8,357.80	￥33,431.20	
10	2012-1-12	Y470N-I	5	￥5,500	0.95	￥1,375.00	￥26,125.00	
11	2012-1-12	ThinkPa	6	￥4,799	0.85	￥4,319.10	￥24,474.90	
12	2012-1-13	G470AH	7	￥3,849	0.75	￥6,735.75	￥20,207.25	
13	2012-1-13	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
14	2012-1-13	ThinkPa	12	￥4,600	0.75	￥13,800.00	￥41,400.00	
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25	
16	2012-1-14	Y470N-I	13	￥5,500	0.9	￥7,150.00	￥64,350.00	
17	2012-1-14	ThinkPa	6	￥4,799	0.8	￥5,758.80	￥23,035.20	
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85	
19	2012-1-15	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20	
20	2012-1-15	ThinkPa	14	￥4,600	0.75	￥16,100.00	￥48,300.00	
21								
22								
23			数量	折扣率				
24			>=8	>=0.85				
25								

高级筛选 － 列表区域：

Sheet1!$A$2:$G$20

Sheet1　Sheet2　Sheet3

图 14-70　选择列表区域

03 单击【列表区域】右侧的展开按钮，展开【高级筛选】对话框，在【列表区域】文本框中显示了所选区域的绝对引用，然后单击【条件区域】文本框右侧的折叠按钮，选择创建的条件区域 C23:D24，如图 14-71 所示。

	A	B	C	D	E	F	G
13	2012-1-13	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20
14	2012-1-13	ThinkPa	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	2012-1-14	Y470N-I	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	2012-1-14	ThinkPa	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	2012-1-15	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20
20	2012-1-15	ThinkPa	14	￥4,600	0.75	￥16,100.00	￥48,300.00
21							
22							
23			数量	折扣率			
24			>=8	>=0.85			
25							
26							

高级筛选 - 条件区域：
Sheet1!Criteria

图 14-71　选择条件区域

04 选择完毕后，单击【条件区域】右侧的展开按钮，展开【高级筛选】对话框，在【条件区域】文本框中显示选择条件区域的绝对引用，如图 14-72 所示。

	A	B	C	D	E	F	G
13	2012-1-13	Y470N-I	8	￥5,499	0.85	￥6,598.80	￥37,393.20
14	2012-1-13	ThinkPa	12	￥4,600	0.75	￥13,800.00	￥41,400.00
15	2012-1-14	G470AH	9	￥3,799	0.75	￥8,547.75	￥25,643.25
16	2012-1-14	Y470N-I	13	￥5,500	0.9	￥7,150.00	￥64,350.00
17	2012-1-14	ThinkPa	6	￥4,799	0.8	￥5,758.80	￥23,035.20
18	2012-1-15	G470AH	7	￥3,849	0.95	￥1,347.15	￥25,595.85
19	2012-1-15	Y470N-I	8	￥5,499			
20	2012-1-15	ThinkPa	14	￥4,600			
21							
22							
23			数量	折扣率			
24			>=8	>=0.85			
25							
26							
27							
28							
29							
30							
31							

高级筛选
方式
◉ 在原有区域显示筛选结果(F)
○ 将筛选结果复制到其他位置(O)
列表区域(L)：Sheet1!$A$2:$G$20
条件区域(C)：Sheet1!Criteria
复制到(T)：
☐ 选择不重复的记录(R)
确定　取消

图 14-72　显示条件区域

05 单击【确定】按钮，得到高级筛选的结果。

【提示】用户如果要撤销筛选出的数据结果，选择【数据】选项卡，单击【排序和筛选】选项组中的【筛选】按钮，即可撤销筛选显示全部数据。



14.3 分析销售结果

商品销售盈利与否，需要通过分析商品销售数据的统计结果，才能得以判定。Excel VBA 是企业进行商品销售数据结果分析的一种有效工具。

14.3.1 销售业绩评价

通过销售额可以评定销售业绩的好坏。销售额的统计比较麻烦，很多企业依靠多劳多得

的政策提高员工工作积极性与热情，一般会通过销售额对员工进行业绩评价，然后再根据评价结果确定其奖金提成和年终奖等。

使用 Excel VBA 进行销售业绩评价，不仅可以避免在 Excel 工作表中计算业绩奖金的麻烦，而且还可以极大地提高评价效率和准确度。

以如图 14-73 所示的笔记本销售数据表为例，使用 Excel VBA 对员工销售业绩进行评价。

笔记本销售数据表

销售人员	产品型号	数量	销售额	业绩评价	业绩奖金
刘增	G470AH	20	￥82,400.00		
张云	Y470N-IFI(H)	14	￥52,000.00		
张军	ThinkPad E420	18	￥72,000.00		
郑光	G470AH	13	￥50,000.00		
张红	Y470N-IFI(H)	10	￥43,000.00		
孔玲	ThinkPad E420	21	￥85,000.00		
王杰	G470AH	11	￥48,700.00		
张磊	G470AH	17	￥68,800.00		
崔艳	Y470N-IFI(H)	13	￥59,200.00		
薛凯	Y470N-IFI(H)	9	￥37,600.00		
李立威	ThinkPad E420	23	￥92,000.00		
李丽	ThinkPad E420	12	￥56,000.00		
韩少华	G470AH	19	￥77,500.00		

销售额标准	业绩评价	奖金提成
40000以下	差	1.00%
40000～50000	一般	2.00%
50000～60000	好	4.00%
60000～70000	较好	6.00%
70000以上	很好	10.00%

Sheet1　Sheet2　Sheet3

图 14-73　商品销售数据表

具体操作步骤如下。

01 在工作表中按 Alt+F11 组合键打开 VBA 代码窗口，从中插入一个模块，并在模块代码窗口中输入下面的“业绩评价”子过程程序代码，如图 14-74 所示。

```
   Public Sub 业绩评价()
   Dim i As Integer '定义一个用于循环的整型变量
   Dim count As Integer  '定义一个用于保存记录数的整型变量
   count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count  '把工作表的记录数赋给
'count
   For i = 3 To count
      Select Case Cells(i, 4)
         Case 0 To 40000
             Cells(i, 5) = "差"       '销售额为 0～40000，则业绩评价为“差”
         Case 40000 To 50000
             Cells(i, 5) = "一般"     '销售额为 40000～50000，则业绩评价为“一般”
         Case 50000 To 60000
             Cells(i, 5) = "好"       '销售额为 50000～60000，则业绩评价为“好”
         Case 60000 To 70000
             Cells(i, 5) = "较好"     '销售额为 60000～70000，则业绩评价为“较好”
         Case 70000 To 999999999
             Cells(i, 5) = "很好"     '销售额为 70000～999999999，则业绩评价为“很好”
      End Select
   Next i
   End Sub
```

```
(通用)                                   业绩评价
Public Sub 业绩评价()
Dim i As Integer '定义一个用于循环的整型变量
Dim count As Integer  '定义一个用于保存记录数的整型变量
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count  '把工作表的记录数赋给cour
For i = 3 To count
   Select Case Cells(i, 4)
      Case 0 To 40000
         Cells(i, 5) = "差"   '销售额在0~40000，则业绩评价为"差"
      Case 40000 To 50000
         Cells(i, 5) = "一般" '销售额在40000~50000，则业绩评价为"一般"
      Case 50000 To 60000
         Cells(i, 5) = "好"    '销售额在50000~60000，则业绩评价为"好"
      Case 60000 To 70000
         Cells(i, 5) = "较好"  '销售额在60000~70000，则业绩评价为"较好"
      Case 70000 To 999999999
         Cells(i, 5) = "很好"    '销售额在70000~999999999，则业绩评价为"很好"
   End Select
Next i
End Sub
```

图 14-74 “业绩评介”子过程代码

02 代码输入完毕后，在菜单栏选择【运行】➤【运行子过程/用户窗体】菜单命令，得到该数据表中每位员工的业绩评价结果，如图 14-75 所示。

笔记本销售数据表

销售人员	产品型号	数量	销售额	业绩评价	业绩奖金
刘增	G470AH	20	￥82,400.00	很好	
张云	Y470N-IFI(H)	14	￥52,000.00	好	
张军	ThinkPad E420	18	￥72,000.00	很好	
郑光	G470AH	13	￥50,000.00	一般	
张红	Y470N-IFI(H)	10	￥43,000.00	一般	
孔玲	ThinkPad E420	21	￥85,000.00	很好	
王杰	G470AH	11	￥48,700.00	一般	
张磊	G470AH	17	￥68,800.00	较好	
崔艳	Y470N-IFI(H)	13	￥59,200.00	好	
薛凯	Y470N-IFI(H)	9	￥37,600.00	差	
李立威	ThinkPad E420	23	￥92,000.00	很好	
李丽	ThinkPad E420	12	￥56,000.00	好	
韩少华	G470AH	19	￥77,500.00	很好	

销售额标准	业绩评价	奖金提成
40000以下	差	1.00%
40000~50000	一般	2.00%
50000~60000	好	4.00%
60000~70000	较好	6.00%
70000以上	很好	10.00%

Sheet1 Sheet2 Sheet3

图 14-75 业绩评价结果

03 为了计算每位员工的“业绩奖金”，需要在“业绩评价”子过程代码框架中添加下面的代码，如图 14-76 所示。

```
Public Sub 业绩评价()
…
For i = 3 To count
   Select Case Cells(i, 4)
      Case 0 To 500000
         ……
         Cells(i, 6) = Cells(i, 4) * 0.01  '业绩评价为“差”时，奖金提成为 1%
         ……
         Cells(i, 6) = Cells(i, 4) * 0.02  '业绩评价为“一般”时，奖金提成为 2%
         ……
         Cells(i, 6) = Cells(i, 4) * 0.04  '业绩评价为“好”时，奖金提成为 4%
         ……
         Cells(i, 6) = Cells(i, 4) * 0.06  '业绩评价为“较好”时，奖金提成为 6%
         ……
```

```
            Cells(i, 6) = Cells(i, 4) * 0.1  '业绩评价为“很好”时，奖金提成为10%
        End Select
    Next i
    End Sub
```

```
(通用)                                    业绩评价
Public Sub 业绩评价()
Dim i As Integer '定义一个用于循环的整型变量
Dim count As Integer  '定义一个用于保存记录数的整型变量
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count  '把工作表的记录数赋给cour
For i = 3 To count
    Select Case Cells(i, 4)
        Case 0 To 40000
            Cells(i, 5) = "差"  '销售额在0~40000，则业绩评价为"差"
            Cells(i, 6) = Cells(i, 4) * 0.01  '业绩评价为"差"时，奖金提成为1%
        Case 40000 To 50000
            Cells(i, 5) = "一般" '销售额在40000~50000，则业绩评价为"一般"
            Cells(i, 6) = Cells(i, 4) * 0.02  '业绩评价为"一般"时，奖金提成为2%
        Case 50000 To 60000
            Cells(i, 5) = "好"    '销售额在50000~60000，则业绩评价为"好"
            Cells(i, 6) = Cells(i, 4) * 0.04  '业绩评价为"好"时，奖金提成为4%
        Case 60000 To 70000
            Cells(i, 5) = "较好"  '销售额在60000~70000，则业绩评价为"较好"
            Cells(i, 6) = Cells(i, 4) * 0.06  '业绩评价为"较好"时，奖金提成为6%
        Case 70000 To 999999999
            Cells(i, 5) = "很好"    '销售额在70000~999999999，则业绩评价为"很好"
            Cells(i, 6) = Cells(i, 4) * 0.1  '业绩评价为"很好"时，奖金提成为10%
    End Select
Next i
End Sub
```

图 14-76　“业绩评价”子过程的完整代码

04 代码输入完毕后，在菜单栏选择【运行】➢【运行子过程/用户窗体】菜单命令，得到业绩奖金的计算结果，如图 14-77 所示。

笔记本销售数据表

销售人员	产品型号	数量	销售额	业绩评价	业绩奖金
刘增	G470AH	20	￥82,400.00	很好	8240
张云	Y470N-IFI(H)	14	￥52,000.00	好	2080
张军	ThinkPad E420	18	￥72,000.00	很好	7200
郑光	G470AH	13	￥50,000.00	一般	1000
张红	Y470N-IFI(H)	10	￥43,000.00	一般	860
孔玲	ThinkPad E420	21	￥85,000.00	很好	8500
王杰	G470AH	11	￥48,700.00	一般	974
张磊	G470AH	17	￥68,800.00	较好	4128
崔艳	Y470N-IFI(H)	13	￥59,200.00	好	2368
薛凯	Y470N-IFI(H)	9	￥37,600.00	差	376
李立威	ThinkPad E420	23	￥92,000.00	很好	9200
李丽	ThinkPad E420	12	￥56,000.00	好	2240
韩少华	G470AH	19	￥77,500.00	很好	7750

销售额标准	业绩评价	奖金提成
40000以下	差	1.00%
40000~50000	一般	2.00%
50000~60000	好	4.00%
60000~70000	较好	6.00%
70000以上	很好	10.00%

Sheet1　Sheet2　Sheet3

图 14-77　业绩奖金计算结果

到此，已经完成了对笔记本销售数据表的销售业绩评价。这里只是提供了一种业绩评价和计算业绩奖金的方法，在实际工作中，用户只需将“销售额”或“业绩奖金”转换成不同的列字段进行业绩评价即可。

14.3.2　利用统计函数汇总分析销售数据

使用统计函数可以方便地从大量销售数据中获取有价值的汇总信息，如统计销售人员数

目、汇总分析销售额等。以图 14-78 为例，使用 Excel VBA 汇总分析销售数据的具体操作内容如下。

笔记本销售额记录表

销售人员	销售额	销售人员	销售额	销售人员	销售额
刘增	￥82,400.00	魏凯	￥82,400.00	王强	￥82,400.00
张云	￥52,000.00	崔占利	￥72,000.00	张明远	￥52,000.00
张军	￥72,000.00	王国博	￥50,000.00	杜森	￥72,000.00
郑光	￥50,000.00	崔晓燕	￥43,000.00	秦明明	￥50,000.00
张红	￥43,000.00	王艳玲	￥85,000.00	王翔	￥43,000.00
孔玲	￥85,000.00	丁玉	￥59,200.00	李佳佳	￥85,000.00
王杰	￥48,700.00	董斌	￥37,600.00	程晓静	￥48,700.00
张磊	￥68,800.00	黄艳玲	￥92,000.00	李静	￥68,800.00
崔艳	￥59,200.00	陈晓娟	￥56,000.00	宋丽霞	￥48,700.00
薛凯	￥37,600.00	张涛涛	￥37,600.00	叶欣	￥68,800.00
李立威	￥92,000.00	陈磊	￥92,000.00	李雅欣	￥37,600.00
李丽	￥56,000.00	张慧杰	￥56,000.00	种源	￥56,000.00
韩少华	￥77,500.00	张克歌	￥77,500.00	郑小波	￥77,500.00

Sheet1 Sheet2 Sheet3

图 14-78　商品销售额记录表

1. 统计销售人员数目

专做市场销售的企业会有大批的销售人员，而且这些人的变更很频繁，管理人员不可能时刻掌握销售人员的数目。可以使用以下 Excel VBA 程序代码实现统计销售人员数目的功能，具体操作步骤如下。

01 打开“笔记本销售额记录表”，按 Alt+F11 组合键打开 VBA 代码窗口，从中插入一个模块，并在模块代码窗口中添加一个“笔记本销售数据汇总分析”公有子过程，然后在该过程框架中输入以下程序代码，如图 14-79 所示。

```
    Public Sub 销售数据汇总分析()
    Dim i, j As Integer           '定义用于循环的整型变量
    Dim temp As Integer          '定义用于记录销售人员数目的整型变量
    Dim count As Integer         '定义用于记录工作表的记录数的整型变量
    Dim TempMsgbox As VbMsgBoxResult      '定义一个消息对话框的返回值的类型
    count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count      '把工作表的记录数赋
'给 count
    For i = 1 To 6
       If Cells(2, i) = "销售额" Then        '判断列标志是不是“销售额”
          For j = 3 To count
             If Cells(j, i) <> "" Then  '判断单元格是否为空
                temp = temp + 1  '累加求和
             End If
          Next j
         End If
    Next i
           TempMsgbox = MsgBox("目前共有销售人员" & temp & "个", vbOKOnly, "销售数据汇总分析")
           ' 将计算结果以对话框的形式输出
    End Sub
```

```
(通用)                                    销售数据汇总分析
Public Sub 销售数据汇总分析()
Dim i, j As Integer              '定义用于循环的整型变量
Dim temp As Integer            '定义用于记录销售人员数目的整型变量
Dim count As Integer           '定义用于记录工作表的记录数的整型变量
Dim TempMsgbox As VbMsgBoxResult        '定义一个消息对话框的返回值的类型
count = Sheets("Sheet1").[A1].CurrentRegion.Rows.count      '把工作表的记录数赋给
For i = 1 To 6
   If Cells(2, i) = "销售额" Then        '判断列标志是不是"销售额"
      For j = 3 To count
         If Cells(j, i) <> "" Then  '判断单元格是否为空
            temp = temp + 1  '累加求和
         End If
      Next j
    End If
Next i
TempMsgbox = MsgBox("目前共有销售人员" & temp & "个", vbOKOnly, "销售数据汇总分
End Sub
```

图 14-79　“销售数据汇总分析”子过程代码

02 代码输入完毕后，选择【运行】➤【选择子过程/用户窗体】菜单命令，即可看到系统自动返回的消息对话框，如图 14-80 所示。

	A	B	C	D	E	F
1	笔记本销售额记录表					
2	销售人员	销售额	销售人员	销售额	销售人员	销售额
3	刘增	￥82,400.00	魏凯	￥82,400.00	王强	￥82,400.00
4	张云	￥52,000.00	崔占利	￥72,000.00	张明远	￥52,000.00
5	张军	￥72,000.00	王国[illegible]	[illegible]	杜森	￥72,000.00
6	郑光	￥50,000.00	崔晓[illegible]	[illegible]	秦明明	￥50,000.00
7	张红	￥43,000.00	王艳[illegible]	[illegible]	王翔	￥43,000.00
8	孔玲	￥85,000.00	丁玉	[illegible]	李佳佳	￥85,000.00
9	王杰	￥48,700.00	董斌	[illegible]	程晓静	￥48,700.00
10	张磊	￥68,800.00	黄艳[illegible]	[illegible]	李静	￥68,800.00
11	崔艳	￥59,200.00	陈晓娟	￥56,000.00	宋丽霞	￥48,700.00
12	薛凯	￥37,600.00	张涛涛	￥37,600.00	叶欣	￥68,800.00
13	李立威	￥92,000.00	陈磊	￥92,000.00	李雅欣	￥37,600.00
14	李丽	￥56,000.00	张慧杰	￥56,000.00	种源	￥56,000.00
15	韩少华	￥77,500.00	张克歌	￥77,500.00	郑小波	￥77,500.00
16						

销售数据汇总分析

目前共有销售人员39个

确定

图 14-80　销售人员数目提示对话框

2. 对销售额进行汇总分析

使用 Excel VBA 对销售额进行汇总分析的具体操作步骤如下。

01 在“销售数据汇总分析”子过程的代码窗口中补充以下代码，如图 14-81 所示。

```
Public Sub 销售数据汇总分析()
…
Dim value(59) As Double   '定义用于记录销售额的双精度型数组
…
For i = 1 To 6
           …
            value(temp) = Cells(j, i)   '用数组记录每个销售人员的销售额
           temp = temp + 1  '累加求和
        …
MsgBox "其中共有" & WorksheetFunction.CountIf(Range("A2:F15"), ">60000") _
& "人销售额大于 60000" & Chr(13) _
& "    共有" & WorksheetFunction.CountIf(Range("A2:F15"), "<40000") _
& "人销售额小于 40000", vbOKOnly, "销售数据汇总分析"
      '在消息对话框中输出销售额的大概情况
```

```
    MsgBox "销售额第一名是" & WorksheetFunction.Large(value(), 1) & "元人民币"
    & Chr(13) & "销售额第二名是" & WorksheetFunction.Large(value(), 2) & "元人民币"
    & Chr(13) & "销售额第三名是" & WorksheetFunction.Large(value(), 3) & "元人民币" _
    & Chr(13) & "销售额最差的是" & WorksheetFunction.Min(value()) & "元人民币" _
    & Chr(13) & "平均销售额是" & WorksheetFunction.Average(value()) & "元人民币" _
    , vbOKOnly, "销售数据汇总分析"
         '在消息对话框中输出销售额的汇总分析情况
    End Sub
```

图 14-81　销售数据汇总分析完整代码

【提示】在这段代码中，首先引入一个用于记录销售额的双精度数组 value ()，在对销售人员的数目进行累加求和时，分别将每人的销售额记录在数组内；然后在 MsgBox 函数内使用统计函数 CountIf () 返回满足条件的销售人员数目；最后在 MsgBox 函数内使用统计函数 Large ()、Min () 和 Average () 分别返回销售额的汇总分析情况。

01 代码输入完毕后，在菜单栏选择【运行】➤【选择子过程/用户窗体】菜单命令，弹出的消息对话框，如图 14-82 所示。

图 14-82　返回总体情况

02 单击【确定】按钮，系统将自动弹出销售额的汇总分析结果，如图 14-83 所示。

	A	B	C	D	E	F
4	张云	￥52,000.00	崔占利	￥72,000.00	张明远	￥52,000.00
5	张军	￥72,000.00	王国博	￥50,000.00	杜森	￥72,000.00
6	郑光	￥50,000.0				￥50,000.00
7	张红	￥43,000.				￥43,000.00
8	孔玲	￥85,000.				￥85,000.00
9	王杰	￥48,700.				￥48,700.00
10	张磊	￥68,800.				￥68,800.00
11	崔艳	￥59,200.				￥48,700.00
12	薛凯	￥37,600.				￥68,800.00
13	李立威	￥92,000.				￥37,600.00
14	李丽	￥56,000.00				￥56,000.00
15	韩少华	￥77,500.00	张克歌	￥77,500.00	郑小波	￥77,500.00

销售数据汇总分析

销售额第一名是92000元人民币
销售额第二名是92000元人民币
销售额第三名是92000元人民币
销售额最差的是0元人民币
平均销售额是40916.6666666667元人民币

确定

图 14-83 返回销售额汇总分析

【提示】 在实际应用中，用户可以根据需要获得相应销售额的汇总分析结果，只需改变 Large(value( ),1) 函数中返回序列数“1”即可。

3. 对销售额进行排序

使用 Excel VBA 程序代码对销售额进行排序，具体操作步骤如下。

01 在上文添加的模块中插入一个“销售排序”公有子过程，并在其中插入以下程序代码，如图 14-84 所示。

```
Public Sub 销售额排位()
Dim Rag As Range                          '定义一个鼠标事件返回值的类型
Dim TempMsgbox As VbMsgBoxResult          '定义一个消息对话框的返回值的类型
Set Rag = Application.Selection              '为鼠标事件赋初值为应用程序的选择事件
If Rag.value = "" Then                    '判断是否选中某个单元格
   TempMsgbox = MsgBox("选择销售员或销售额", vbOKOnly, "销售额排位")
                                          '弹出提示对话框
Else
   If WorksheetFunction.IsNumber(Rag.value) Then   '判断选中的单元格中是否是数字，
'即销售额
      TempMsgbox = MsgBox(Rag.Offset(0, -1).value & "的销售额是第"
  & WorksheetFunction.Rank(Rag.value, Range("A2:F15"), 0) & "位", vbOKOnly, "
销售额排位")
         '如果选中的是销售额，则返回到销售额所在单元格的前一个单元格输出其销售人员姓名
    Else
      If WorksheetFunction.IsNumber(Rag.Offset(0, 1).value)  Then
         TempMsgbox = MsgBox(Rag.value & "的销售额是第" _
      & WorksheetFunction.Rank(Rag.Offset(0, 1).value, Range("A2:F15"), 0) &
"位" _
      , vbOKOnly, "销售额排位")
        '如果选中的是销售人员，则将销售人员的销售额排位后输出
      Else
        TempMsgbox = MsgBox("错误的选择", vbOKOnly, "销售额排位")
          '如果选中的是空的单元格，或其他单元格，则提示“错误信息”
      End If
   End If
End If
End Sub
```

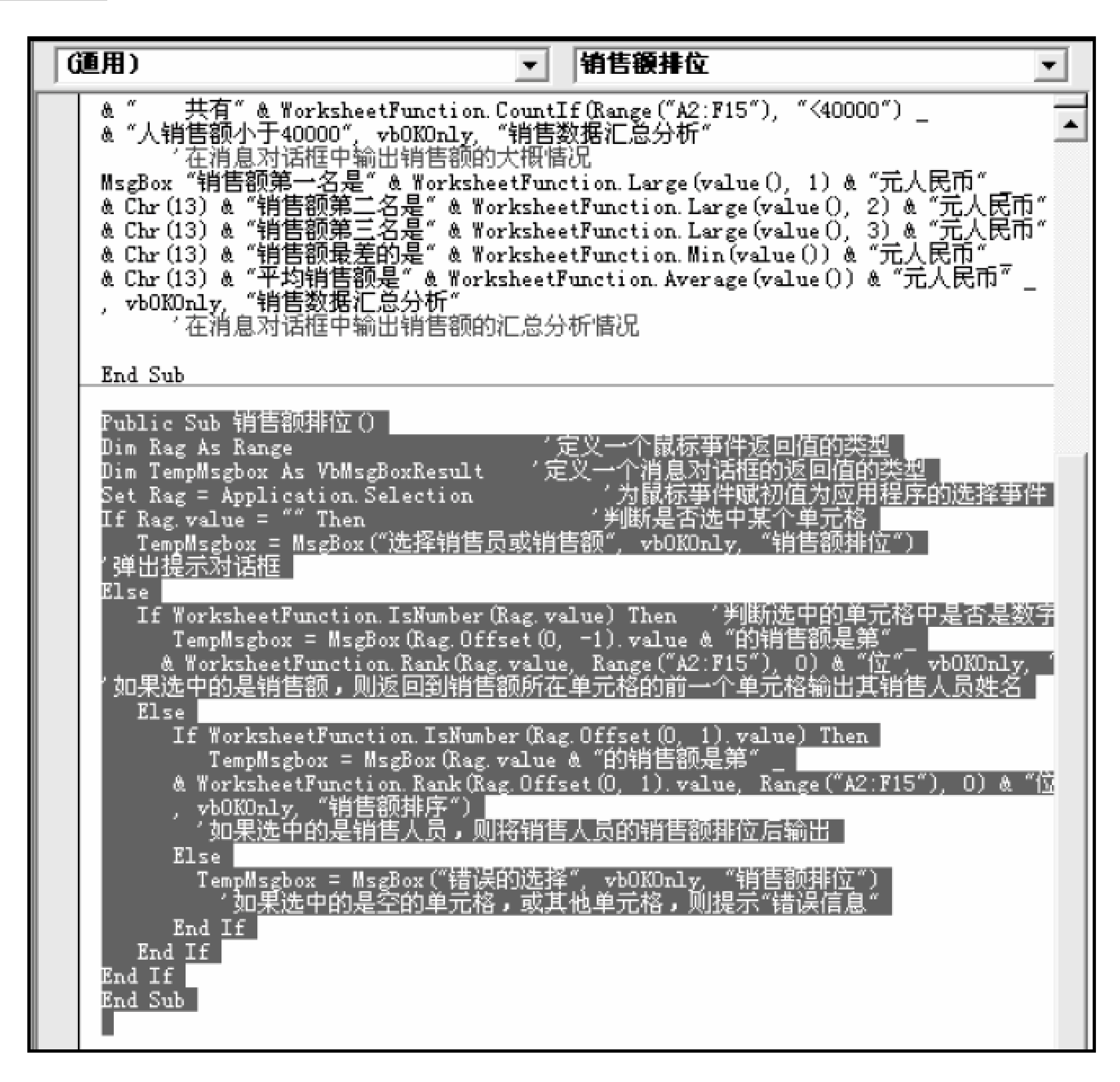


图 14-84　销售额排位子过程代码

【提示】在以上代码中，首先定义一个鼠标事件返回值类型的变量 Rag，并将应用程序的选择事件赋值给 Rag；然后使用 If 语句判断用户是否选择了某个单元格，如果没有，则弹出提示信息，否则继续判断该选中单元格的内容是"销售额"还是"销售员"，并针对不同的单元格内容分别进行排位输出，如果用户选择的是其他无关单元格，则将弹出"错误的选择"提示信息。

【注意】在使用这段代码时，"销售额"字段下的单元格格式不能是"货币"型，可以用"数值"型或"常规"等类型表示。

02 在工作表中为上述程序设置执行按钮。选择功能区【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中单击【按钮】按钮 (窗体控件)，然后在工作表的适当位置拖放产生该按钮，系统会自动弹出【指定宏】对话框，在【宏名】列表框中选择"销售额排位"选项，如图 14-85 所示。



图 14-85　【指定宏】对话框

03 单击【确定】按钮，完成宏的指定操作。然后右击插入的按钮，在弹出的快捷菜单中选择【编辑文字】菜单命令，将按钮名称更改为“销售额排位”，如图 14-86 所示。

04 退出该按钮的编辑状态，单击【销售额排位】按钮，执行指定的过程，系统即可根据用户在工作表中选中的单元格返回相应的消息对话框。如果选中的是空的单元格，则弹出“选择销售员或销售额”的消息对话框，如图 14-87 所示。



	A	B	C	D	E	F
4	张云	¥52,000.00	崔占利	¥72,000.00	张明远	¥52,000.00
5	张军	¥72,000.00	王国博	¥50,000.00	杜森	¥72,000.00
6	郑光	¥50,000.00	崔晓燕	¥43,000.00	秦明明	¥50,000.00
7	张红	¥43,000.00	王艳玲	¥85,000.00	王翔	¥43,000.00
8	孔玲	¥85,000.00	丁玉	¥59,200.00	李佳佳	¥85,000.00
9	王杰	¥48,700.00	董斌	¥37,600.00	程晓静	¥48,700.00
10	张磊	¥68,800.00	黄艳玲	¥92,000.00	李静	¥68,800.00
11	崔艳	¥59,200.00	陈晓娟	¥56,000.00	宋丽霞	¥48,700.00
12	薛凯	¥37,600.00	张涛涛	¥37,600.00	叶欣	¥68,800.00
13	李立威	¥92,000.00	陈磊	¥92,000.00	李雅欣	¥37,600.00
14	李丽	¥56,000.00	张慧杰	¥56,000.00	种源	¥56,000.00
15	韩少华	¥77,500.00	张克歌	¥77,500.00	郑小波	¥77,500.00

图 14-86　更改按钮名称并设置控件格式



图 14-87　选中空的单元格

05 如果选择了类似 B2 等其他无关的单元格，系统则弹出“错误的选择”消息对话框，如图 14-88 所示。

06 如果选择的是“销售人员”列的姓名单元格，系统则弹出如图 14-89 所示的消息对话框。



	A	B	C	D	E	F
1	笔记本销售额记录表					
2	销售人员		销售人员	销售额	销售人员	销售额
3	刘增	¥82,400.00	魏凯	¥82,400.00	王强	¥82,400.00
4	张云	¥52,000.00	崔	000.00	张明远	¥52,000.00
5	张军	¥72,000.00	王	000.00	杜森	¥72,000.00
6	郑光	¥50,000.00	崔	000.00	秦明明	¥50,000.00
7	张红	¥43,000.00	王	000.00	王翔	¥43,000.00
8	孔玲	¥85,000.00	丁	200.00	李佳佳	¥85,000.00
9	王杰	¥48,700.00	董斌	¥37,600.00	程晓静	¥48,700.00
10	张磊	¥68,800.00	黄艳玲	¥92,000.00	李静	¥68,800.00
11	崔艳	¥59,200.00	陈晓娟	¥56,000.00	宋丽霞	¥48,700.00
12	薛凯	¥37,600.00	张涛涛	¥37,600.00	叶欣	¥68,800.00
13	李立威	¥92,000.00	陈磊	¥92,000.00	李雅欣	¥37,600.00
14	李丽	¥56,000.00	张慧杰	¥56,000.00	种源	¥56,000.00
15	韩少华	¥77,500.00	张克歌	¥77,500.00	郑小波	¥77,500.00

图 14-88　错误选择提示信息

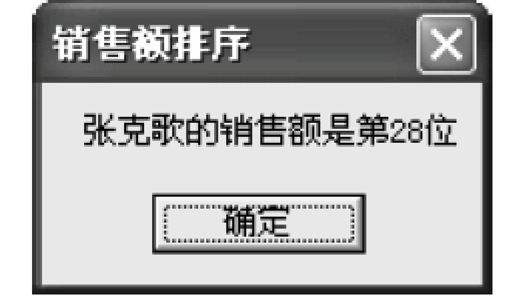


图 14-89　张克歌的销售额排位



14.4 高手私房菜

技巧 1：如何撤销数据筛选操作

执行完数据筛选代码后，自动对数据启用筛选功能，如果想要取消该功能，再次单击筛选功能的执行按钮即可。

技巧 2：快速设置销售金额为货币格式

计算出的金额默认会按照原来单元格的格式显示数据，如果要显示为货币格式的话，可以选中销售金额列的单元格，然后右击，在弹出的快捷菜单中选择【设置单元格格式】菜单命令，打开【设置单元格格式】对话框，在【数字】选项卡中设置单元格格式为“货币”。

第 15 章　用 Excel VBA 实现高效文件处理

使用 Excel 开发的很多应用程序都需要处理外部的文件，只使用 Excel 的内置函数往往达不到这种要求，此时就需要使用 VBA 来解决。

另外，企业管理人员在对多个相关表格文件中的数据进行管理时，经常要在多个文档之间进行切换，这样不仅操作起来比较麻烦，而且也很容易出现错误。本章将使用 Excel VBA 进行高效文件处理操作，不仅可以使企业管理人员的这一工作变得简单方便，而且还可以保证文件数据的安全。



15.1 显示文件列表

为了在 Excel 中显示某个目录下的所有文件列表，可以使用与文件有关的 VBA 函数，也可以使用 FileSearch 对象。

15.1.1 使用 VBA 函数显示文件列表

Dir 函数可以返回与指定的模式或文件属性相匹配的文件名或目录，用户可以使用该函数来显示文件列表。

使用 VBA 函数显示文件列表的具体操作步骤如下。

01 建一张空白工作簿，在 Sheet1 工作表中选择【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中单击 ActiveX 控件组中的【命令按钮】按钮，在 Sheet1 工作表中添加一个命令按钮，并将其 Caption 和"名称"属性值修改为"显示文件列表"，如图 15-1 所示。

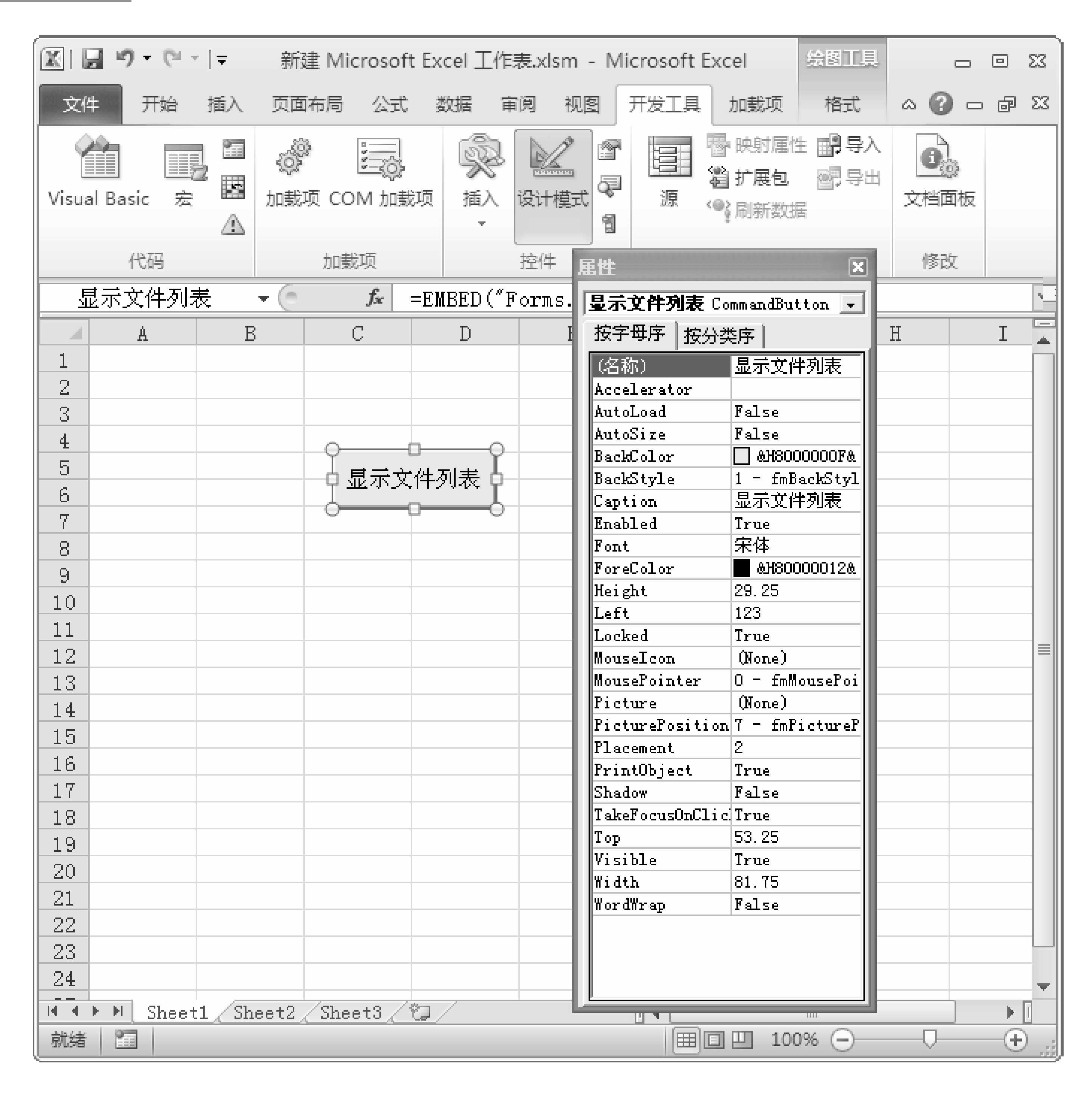


图 15-1　修改命令按钮属性

02 在设计模式下双击插入的【显示文件列表】按钮，进入该按钮的代码编辑窗口，输入下面的代码程序，如图 15-2 所示。

```
Private Sub 显示文件列表_Click()
    Dim message As String
    Dim directory As String, s As String
    Dim row As Long
    Dim fs As Object
    message = "请选择需要显示文件列表的文件路径！"
    directory = GetDirectory(message)   '通过调用函数返回文件夹的路径
    If directory = "" Then
       Exit Sub
    End If
    If Right(directory, 1) <> "\" Then
      directory = directory & "\"   '确定文件夹的路径
    End If
    row = 4    '从工作表的第 4 行开始显示
    Cells.ClearContents   '清除工作表中的原有内容
    '设置标题行的内容、字体
    Cells(row, 1) = "文件名称"
    Cells(row, 2) = "文件类型"
    Cells(row, 3) = "大小"
    Cells(row, 4) = "修改日期"
    Range("A4:D4").Font.Bold = True
    Range("A4:D4").Font.Color = RGB(10, 200, 200)
    s = Dir(directory, 7)      '获取第一个文件
    Do While s <> ""
        row = row + 1
        Cells(row, 1) = s    '文件的名称
        Set fs = CreateObject("Scripting.FileSystemObject")
```

```
            Cells(row, 2) = fs.GetFile(directory & s).Type '文件的类型
            If FileLen(directory & s) = 0 Then  '文件大小
                Cells(row, 3) = "0 KB"
            Else
                Cells(row, 3) = Trim(Str(Int(FileLen(directory & s) / 1024) + 1)) &
" KB"
            End If
            Cells(row, 4) = FileDateTime(directory & s)  '修改时间
            s = Dir    '获取下一个文件
        Loop
        Columns("A:D").EntireColumn.AutoFit  '自动调整列宽

    End Sub
```

这段代码首先使用 GetDirectory 自定义函数获取文件夹的名称及其存放路径；然后自动清空工作表中的原有内容，并设置标题的字体及其格式；再使用 Dir 函数检索文件的名称及其路径，第 1 次使用 Dir 函数的目的是检索找到的第 1 个文件名，第 2 次使用 Dir 函数的目的是检索其他的文件名，当找不到更多的文件时该函数就会自动返回一个空字符串；最后使用 AutoFit 方法自动调整 A~D 列的列宽，使得这 4 列中的数据能够自动地显示在当前单元格中。

代码中的“s = Dir（directory,7）”语句是用来获取文件的显示列表。其中第 1 个参数用于指定文件的存放路径；第 2 个参数用于指定文件的属性，当该参数值为 7 时就可以检索所有的文件，无论是只读文件、隐藏文件，还是系统文件。

代码中还用到与文件有关的其他命令，分别是 FileLen、FileDateTime 和 GetFile。FileLen 返回文件的大小，单位为字节；FileDateTime 返回最后一次修改文件的日期和时间；GetFile 返回一个和指定路径中文件相对应的 File 对象。

代码中用到了 FileSystemObject 对象（简称 FSO 对象），FSO 对象不是 VBA 的一部分，而是以组件的形式提供的。FSO 对象提供大量的方法、属性和事件，可以方便地操作驱动器、文件夹和文件。使用 FSO 对象需要先定义 Object 变量，然后使用 CreateObject 方法创建对象。



图 15-2　输入按钮响应代码

为了实现显示文件列表的功能，还需要再新建一个模块，并在模块代码窗口中添加声明函数的语句、用户自定义数据类型的创建语句。

03 在 VBA 代码窗口中选择【插入】➤【模块】菜单命令，即可插入一个新模块，默认名称为“模块 1”，如图 15-3 所示。

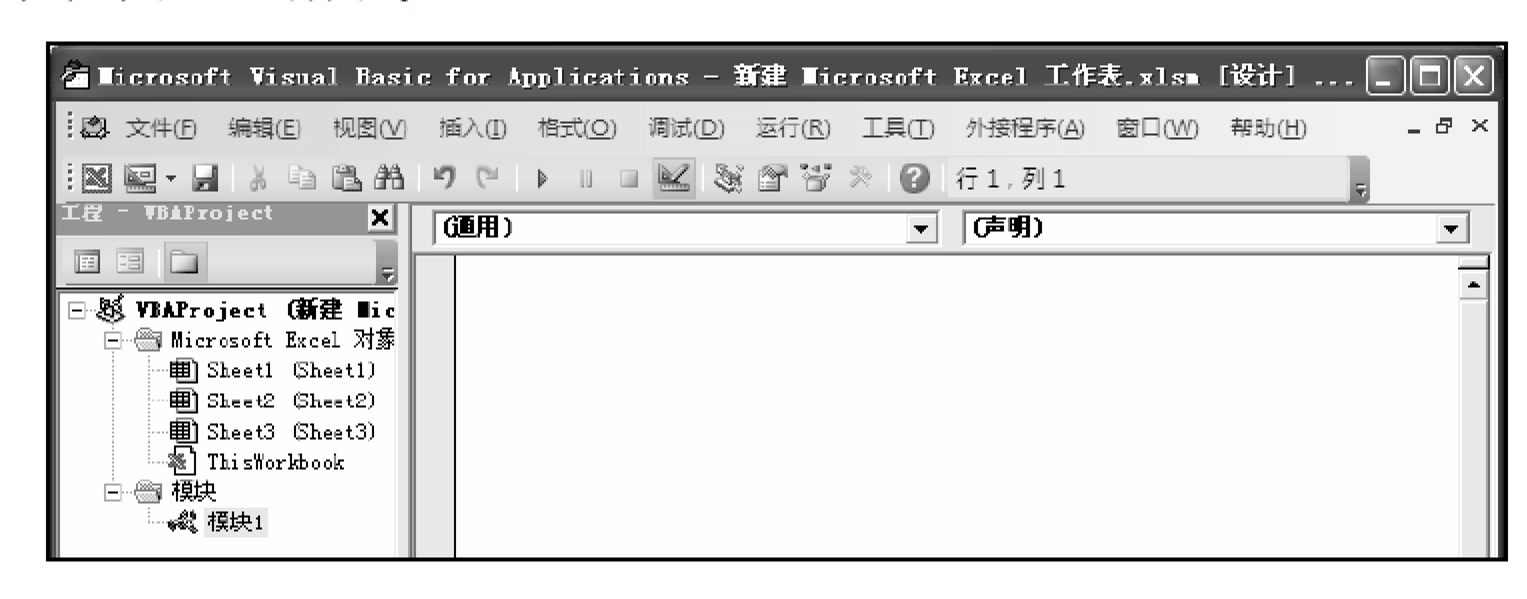


图 15-3 插入模块

04 在“模块 1”代码窗口中输入下面的程序代码，如图 15-4 所示。

```
    Option Explicit '避免有未定义的变量
    Declare Function SHBrowseForFolder Lib "SHELL32.DLL" (lpBrowseInfo As
BROWSEINFO) As Long
    Declare Function SHGetPathFromIDList Lib "SHELL32.DLL" (ByVal pidl As Long,
ByVal pszPath As String) As Long
    Public Type BROWSEINFO      '创建一个用户定义数据类型 BROWSEINFO
        Owner As Long
        Root As Long
        DisplayName As String
        Title As String
        Flags As Long
        lpfn As Long
        Param As Long
        Image As Long
    End Type
```

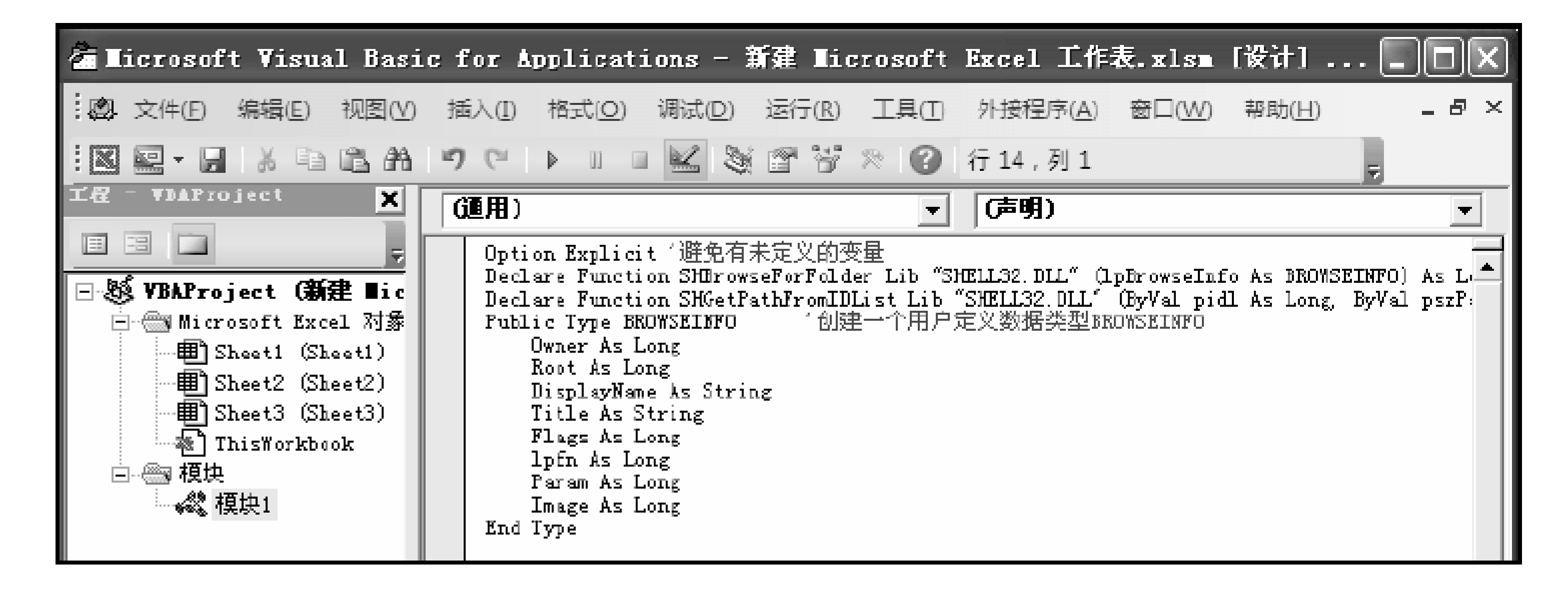


图 15-4 在模块 1 中输入代码

这段代码中声明的 SHGetPathFromIDList 和 SHBrowseForFolder 都是 32 位 Windows API 函数。Windows API（Application Programming Interface，应用程序编程接口）是 Windows 操作系统提供给用户进行系统编程和外设控制的强大函数库，凡是在 Windows 工作环境下执行的应用程序都可以调用，不过使用之前必须进行声明。在 VBA 中声明代码必须位于模块的

顶部，声明之后即可在 VBA 子过程中直接使用。

接下来介绍一下 SHGetPathFromIDList 和 SHBrowseForFolder 这两个函数的作用。SHGetPathFromIDList 的作用是打开【浏览文件夹】对话框，用户可以从中选择文件夹所在的路径；SHBrowseForFolder 的作用是记录用户在该对话框中进行的操作。如果用户单击的是对话框中的【确定】按钮，则该函数的返回值就为 1；如果单击的是【取消】按钮，则该函数的返回值就为 0。

因为 SHBrowseForFolder 应用程序需要处理有关文件夹的信息，所以这里需要创建一个用户自定义数据类型 BROWSEINFO，用来记录有关文件夹的所有资料信息。自定义数据类型也必须位于模块的顶部，即在其他过程之前定义。在创建了用户定义数据类型之后，用户只需使用 Dim 语句将某个变量声明为这种类型，然后使用点运行符“.”引用其中的各个变量即可。

05 在上述代码窗口中继续输入自定义函数 GetDirectory 的代码程序，如图 15-5 所示。

```
Function GetDirectory(Optional message) As String
Dim bInfo As BROWSEINFO      '将变量声明为用户定义数据类型 BROWSEINFO
Dim path As String
Dim r As Long, x As Long, pos As Integer
bInfo.Root = 0            '确定文件夹的根目录是桌面
bInfo.Title = message       '确定【浏览文件夹】对话框中显示的文本
bInfo.Flags = &H1          '确定路径的返回类型
x = SHBrowseForFolder(bInfo)    '弹出【浏览文件夹】对话框
path = Space$(256)           '确定用户选择的文件夹的路径长度
r = SHGetPathFromIDList(ByVal x, ByVal path)  '记录用户在对话框中进行的操作
If r Then                  '得到路径的返回值
    pos = InStr(path, Chr(0))
    GetDirectory = Left(path, pos - 1)
Else
    GetDirectory = ""
End If
End Function
```

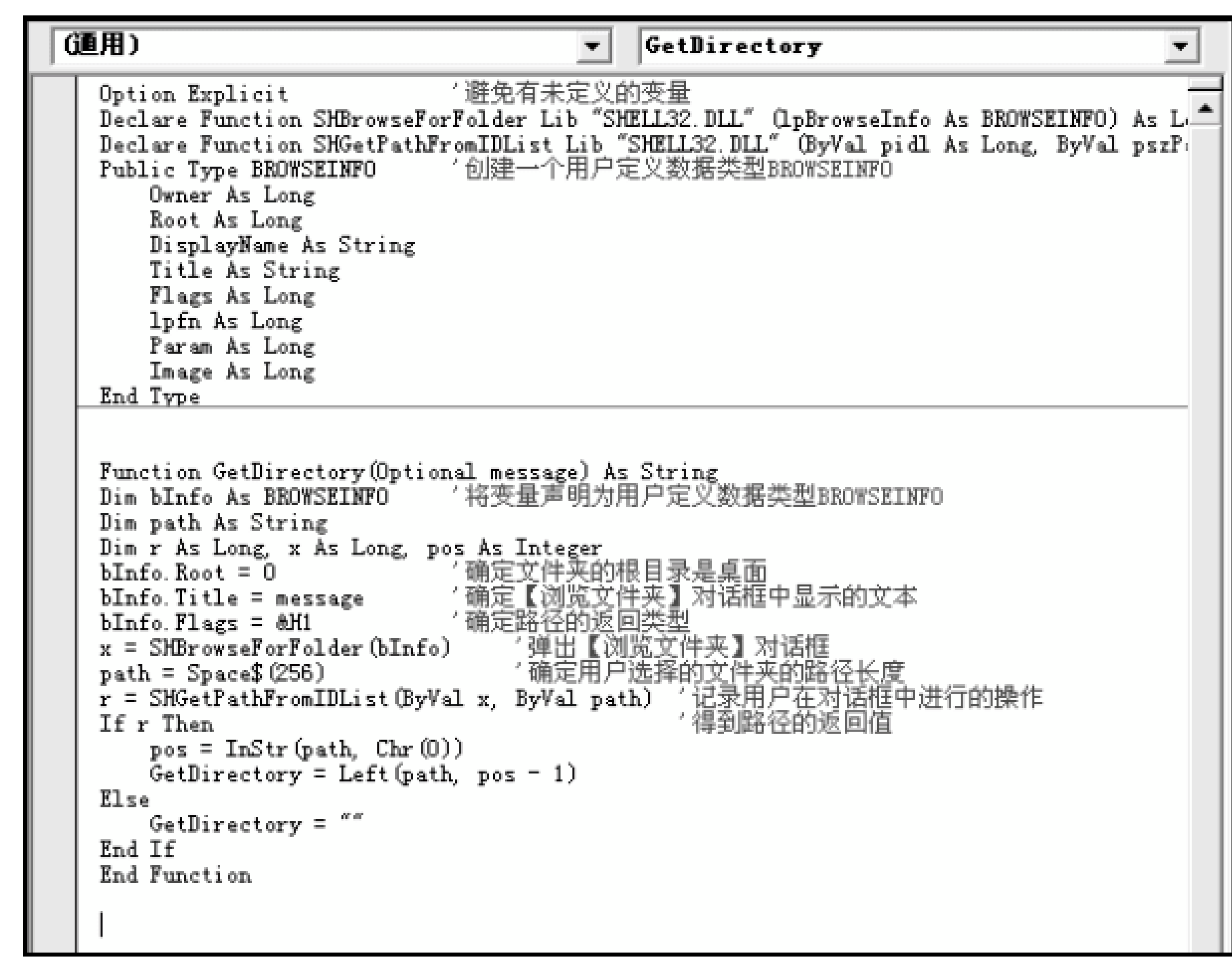


图 15-5　输入自定义函数 GetDirectory 的代码

自定义函数 GetDirectory 的作用是返回在【浏览文件夹】对话框中用户所选择的需要查看的文件夹，然后使用变量 r 记录用户在该对话框中进行的操作，并将用户选择的结果即文件夹的路径返回调用它的显示文件列表_Click（）过程中。

06 代码输入完毕后，关闭 VBA 代码窗口，返回 Sheet1 工作表中，切换到【开发工具】选项卡，在【控件】选项组中单击【退出设计模式】按钮，退出命令按钮的设计模式。然后单击【显示文件列表】按钮，系统就会自动弹出【浏览文件夹】对话框，如图 15-6 所示。

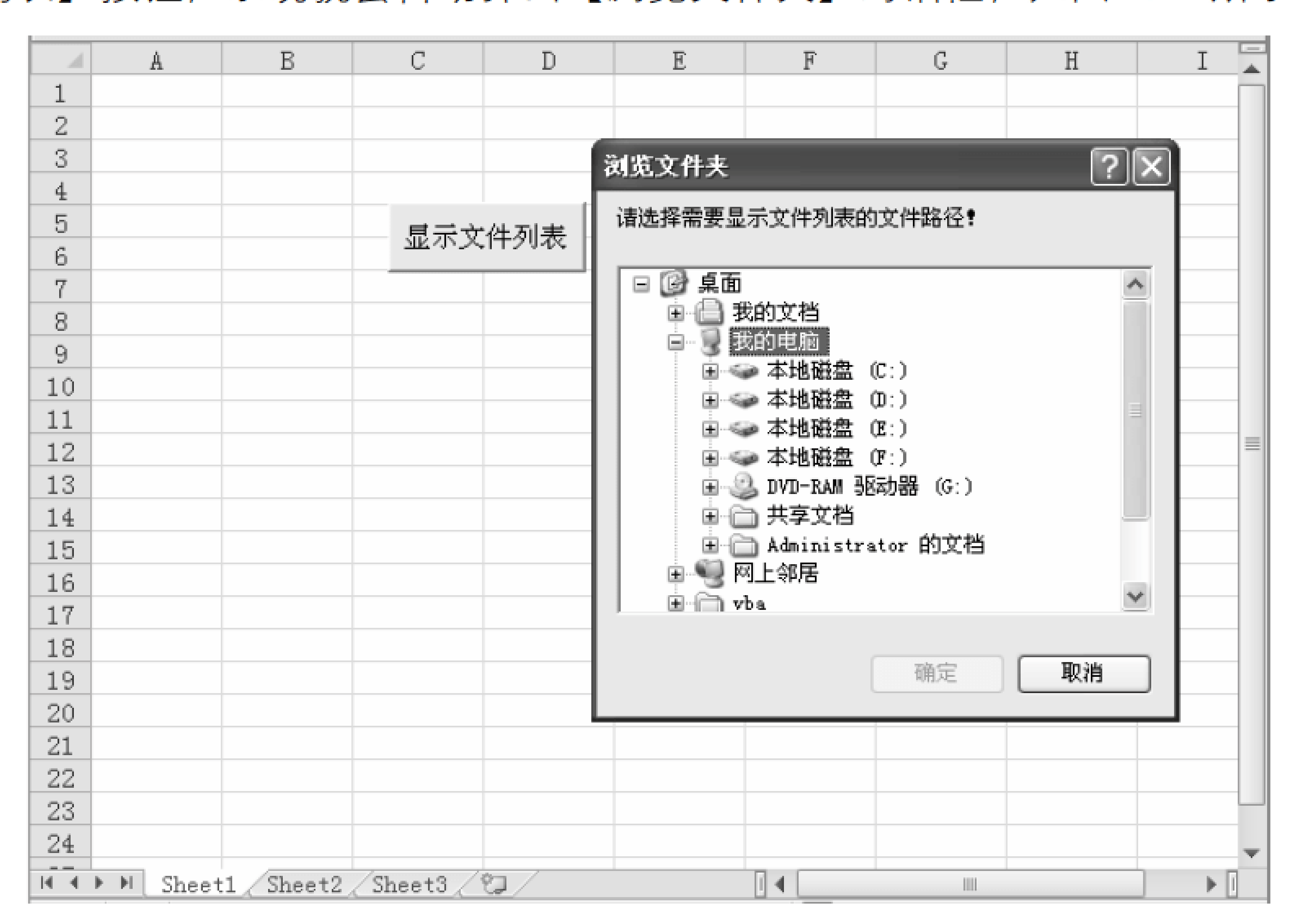


图 15-6 【浏览文件夹】对话框

07 在该对话框的列表框中选择查看的文件夹，如图 15-7 所示。

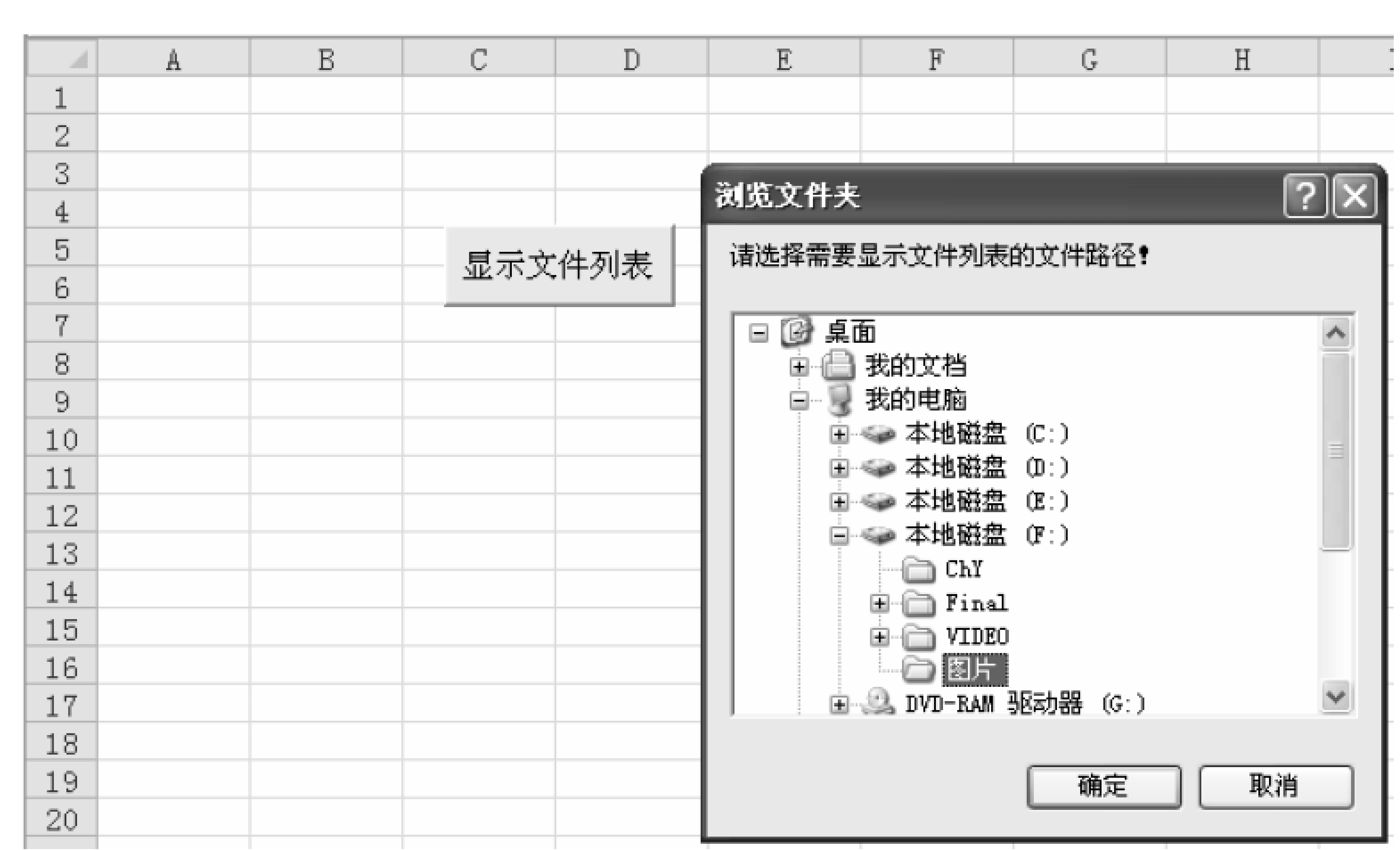


图 15-7 选择要显示文件列表的文件路径

08 选择完毕后，单击【确定】按钮即可显示要查看文件的结果列表，如图 15-8 所示。



文件名称	文件类型	大小	修改日期
Blue hills.jpg	JPEG 图像	28 KB	2008-4-15 9:00
BS9039.jpg	JPEG 图像	21 KB	2005-8-4 21:08
BS9050.jpg	JPEG 图像	20 KB	2005-8-4 21:08
BS9064.jpg	JPEG 图像	24 KB	2005-8-4 21:08
BS9068.jpg	JPEG 图像	19 KB	2005-8-4 21:08
BS9070.jpg	JPEG 图像	23 KB	2005-8-4 21:08
BS9079.jpg	JPEG 图像	25 KB	2005-8-4 21:08
BS9084.jpg	JPEG 图像	27 KB	2005-8-4 21:08
BS9092.jpg	JPEG 图像	22 KB	2005-8-4 21:08
BS9097.jpg	JPEG 图像	24 KB	2005-8-4 21:08
c_1366_768_2736.jpg	JPEG 图像	243 KB	2011-10-15 20:31
Sunset.jpg	JPEG 图像	70 KB	2008-4-15 9:00
Water lilies.jpg	JPEG 图像	82 KB	2008-4-15 9:00
Winter.jpg	JPEG 图像	104 KB	2008-4-15 9:00
xiongy.jpg	JPEG 图像	811 KB	2011-8-20 9:42
未命名.jpg	JPEG 图像	50 KB	2011-8-21 17:02

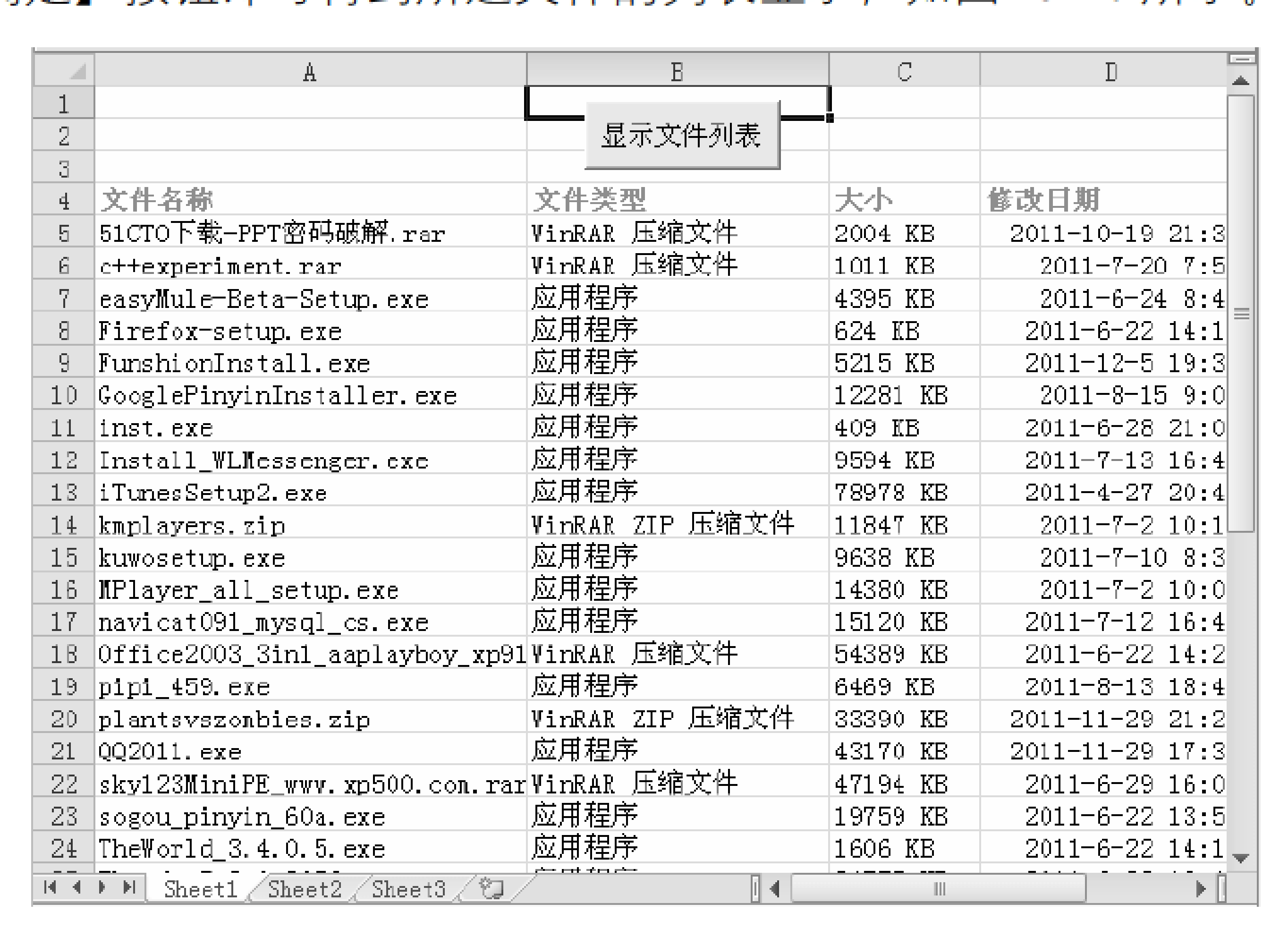
图 15-8　显示要查看的结果列表

09 再一次单击【显示文件列表】按钮，在打开的【浏览文件夹】对话框中选择另一个文件夹，如图 15-9 所示。



图 15-9　选择另一个文件夹

10 单击【确定】按钮即可得到所选文件的列表显示，如图 15-10 所示。



文件名称	文件类型	大小	修改日期
51CTO下载-PPT密码破解.rar	WinRAR 压缩文件	2004 KB	2011-10-19 21:3
c++experiment.rar	WinRAR 压缩文件	1011 KB	2011-7-20 7:5
easyMule-Beta-Setup.exe	应用程序	4395 KB	2011-6-24 8:4
Firefox-setup.exe	应用程序	624 KB	2011-6-22 14:1
FunshionInstall.exe	应用程序	5215 KB	2011-12-5 19:3
GooglePinyinInstaller.exe	应用程序	12281 KB	2011-8-15 9:0
inst.exe	应用程序	409 KB	2011-6-28 21:0
Install_WLMessenger.exe	应用程序	9594 KB	2011-7-13 16:4
iTunesSetup2.exe	应用程序	78978 KB	2011-4-27 20:4
kmplayers.zip	WinRAR ZIP 压缩文件	11847 KB	2011-7-2 10:1
kuwosetup.exe	应用程序	9638 KB	2011-7-10 8:3
MPlayer_all_setup.exe	应用程序	14380 KB	2011-7-2 10:0
navicat091_mysql_cs.exe	应用程序	15120 KB	2011-7-12 16:4
Office2003_3in1_aaplayboy_xp91	WinRAR 压缩文件	54389 KB	2011-6-22 14:2
pipi_459.exe	应用程序	6469 KB	2011-8-13 18:4
plantsvszombies.zip	WinRAR ZIP 压缩文件	33390 KB	2011-11-29 21:2
QQ2011.exe	应用程序	43170 KB	2011-11-29 17:3
sky123MiniPE_www.xp500.com.rar	WinRAR 压缩文件	47194 KB	2011-6-29 16:0
sogou_pinyin_60a.exe	应用程序	19759 KB	2011-6-22 13:5
TheWorld_3.4.0.5.exe	应用程序	1606 KB	2011-6-22 14:1

图 15-10　显示所选文件的列表结果

在两次的结果对比中可以发现，在生成新的文件列表之前系统总是首先删除工作表中的原有记录，然后再将找到的新记录显示在工作表中。

15.1.2 使用 FileDialog 对象显示文件列表

FileDialog 对象是 Microsoft Office 对象中的一个成员，该对象的 Show 方法可以显示并选择文件夹下的文件，使用 AllowMultiSelect 属性可以同时选择多个文件。

使用 FileDialog 对象显示并选择指定文件夹中的文件，具体操作步骤如下。

01 新建一个空白工作簿，在 Sheet1 工作表中切换到【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中单击【按钮】按钮 （窗体控件），在 Sheet1 工作表中添加一个名为"按钮 1"的窗体按钮，此时系统将自动弹出【指定宏】对话框，如图 15-11 所示。



图 15-11 【指定宏】对话框

02 在该对话框中单击【新建】按钮，打开 VBA 代码窗口，此时系统将自动在该项目工程中添加一个名为"模块 1"的模块，并在该模块代码中自动出现"按钮 1"的单击事件处理程序代码框架，如图 15-12 所示。

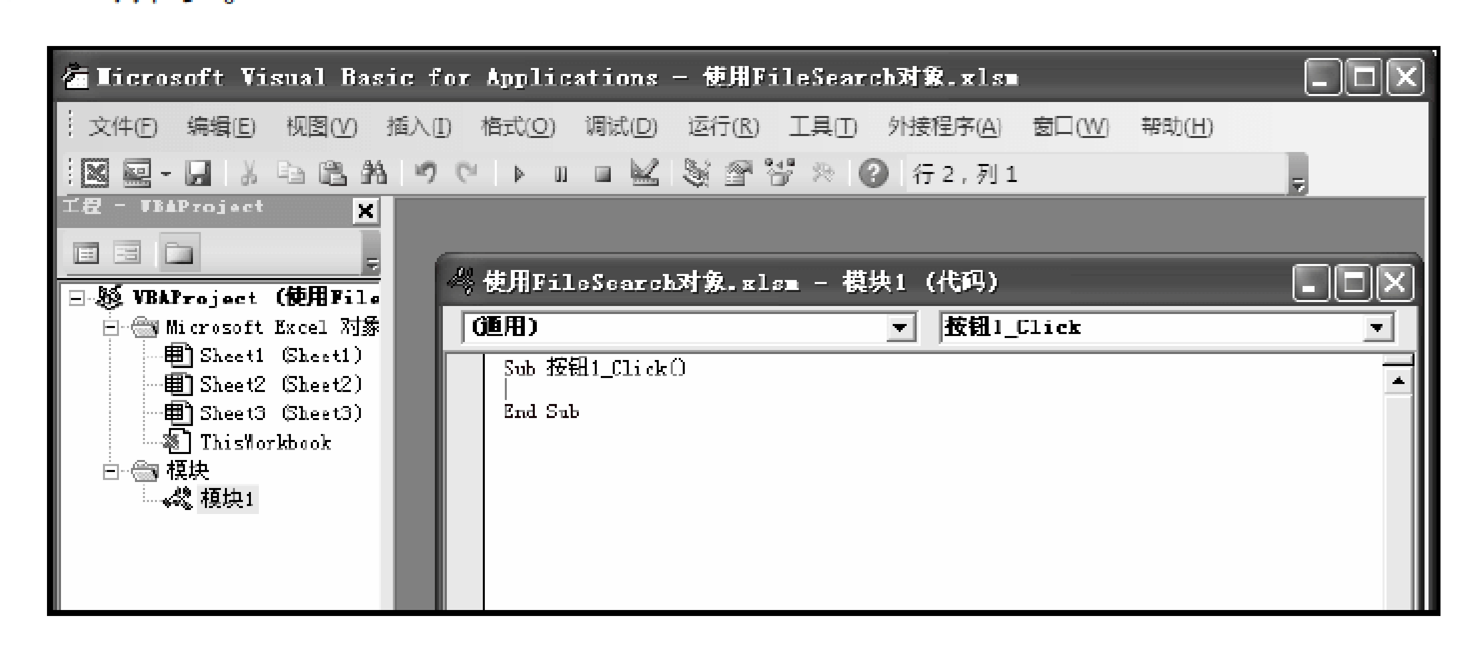


图 15-12 按钮 1 的单击事件处理程序框架

03 在模块代码窗口中添加下面的程序代码。

```
Sub 按钮1_Click()
Dim message As String
```

```
    Dim directory As String, s As String
    Dim row As Long, i As Long
    Dim fs As Object
    row = 4
    Cells.ClearContents         '清除工作表中的原有内容
    Cells(row, 1) = "文件名称"
    Cells(row, 2) = "文件大小"
    Cells(row, 3) = "文件类型"
    Cells(row, 4) = "修改日期"
    Range("A4:D4").Font.Name = "楷体_GB2312"
    Range("A4:D4").Font.Size = 14
    Range("A4:D4").Font.Bold = True
    row = row + 1

    Dim fd As FileDialog
        Set fd = Application.FileDialog(msoFileDialogFilePicker)
        Dim vrtSelectedItem As Variant
        With fd
            .AllowMultiSelect = True
            If .Show = -1 Then
                For Each vrtSelectedItem In .SelectedItems
                    Set fs = CreateObject("Scripting.FileSystemObject")
                    Cells(row, 1) = vrtSelectedItem
                    If FileLen(vrtSelectedItem) = 0 Then  '获取文件大小
                        Cells(row, 2) = "0 KB"
                    Else
                        Cells(row, 2) = Trim(Str(Int(FileLen(vrtSelectedItem) / 1024)
+ 1)) & " KB"
                    End If
                    Cells(row, 3) = fs.GetFile(vrtSelectedItem).Type '获取文件的类型
                    Cells(row, 4) = FileDateTime(vrtSelectedItem)    '获取文件的修改
'日期和时间
                    row = row + 1

                Next vrtSelectedItem
            Else
            End If
        End With
        Set fd = Nothing

    Columns("A:D").EntireColumn.AutoFit
    End Sub
```

04 返回 Sheet1 工作表中，将添加的窗体按钮重命名为“显示文件列表”。然后单击【显示文件列表】按钮，系统就会自动弹出【浏览】对话框，如图 15-13 所示，从中选择要查看的文件，按 Ctrl 键可以同时选中多个文件，如图 15-14 所示。



图 15-13 【浏览】对话框

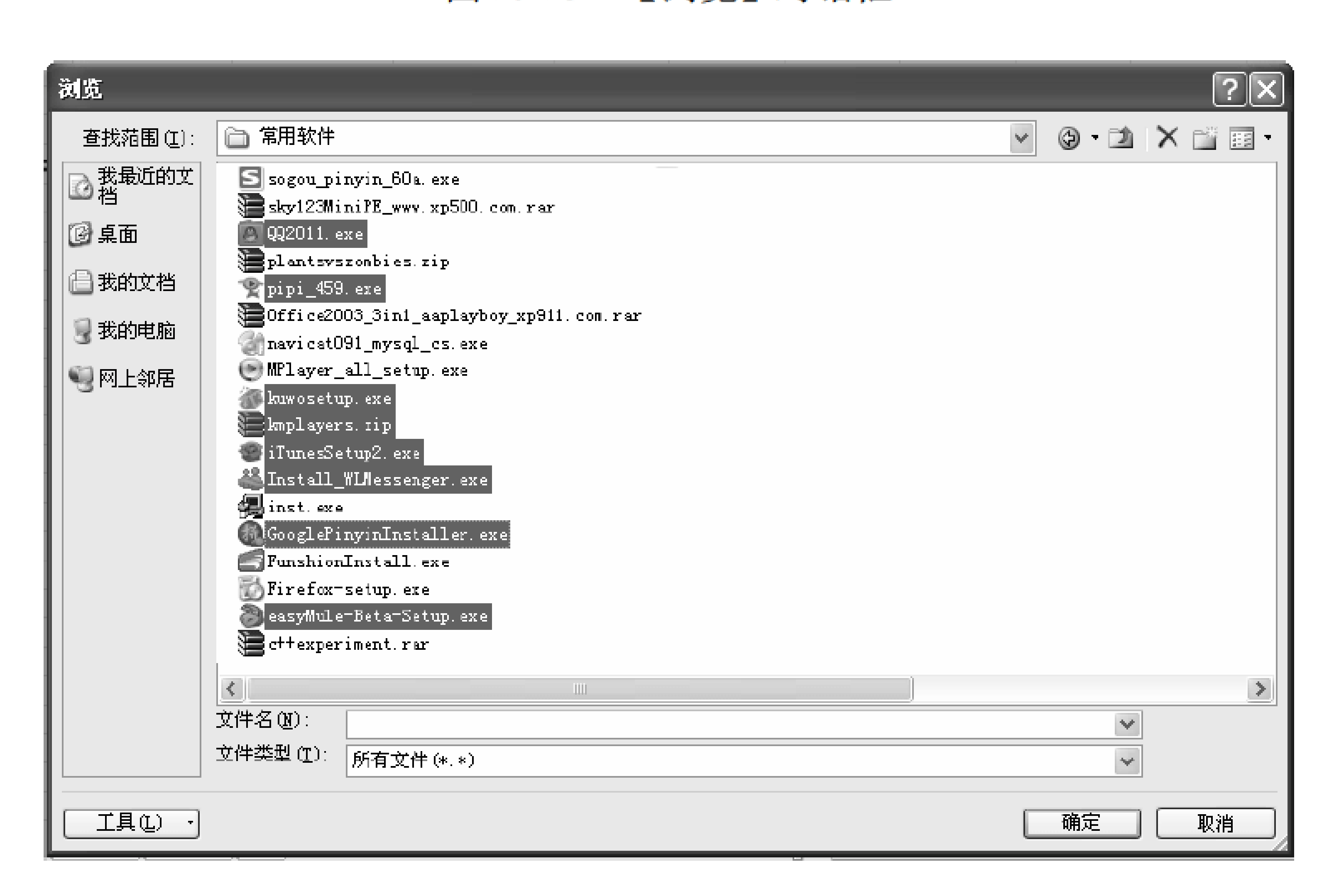
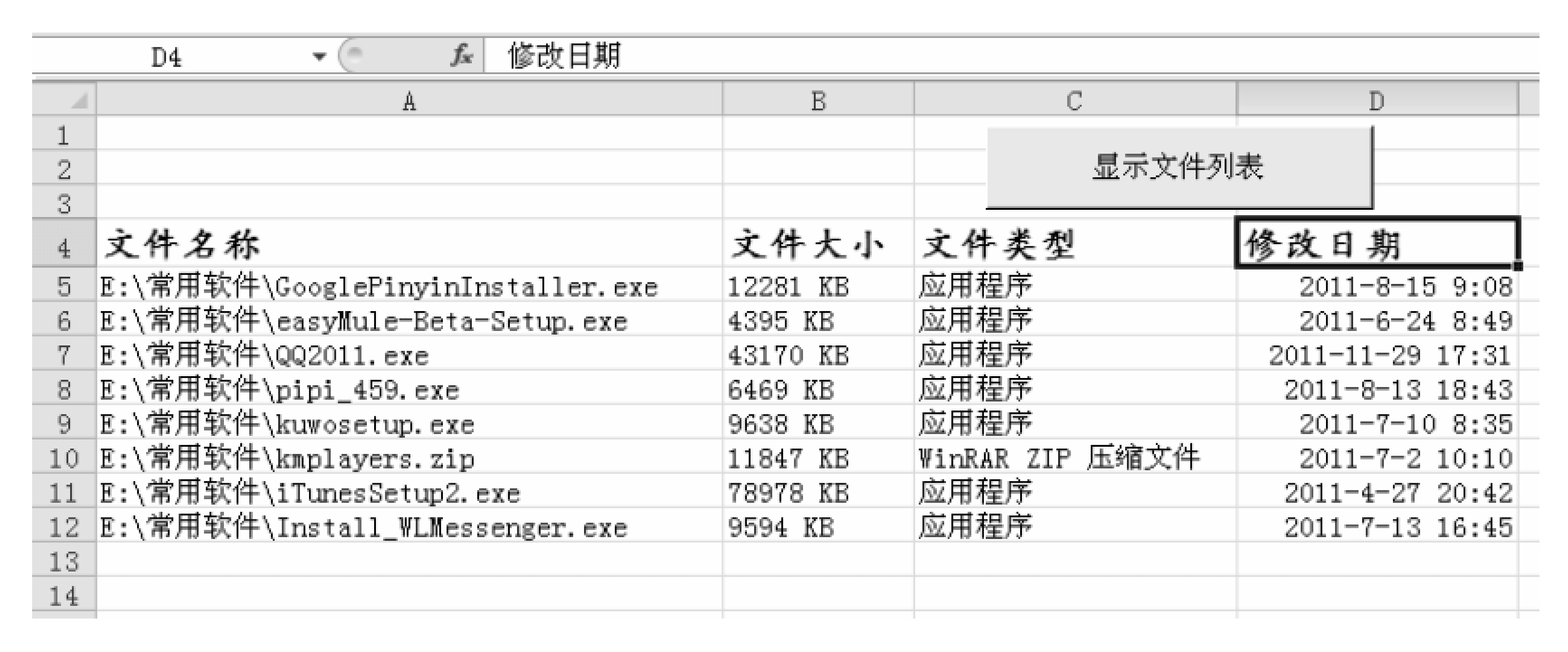


图 15-14 选中要查看的多个文件

05 单击【确定】按钮即可得到所查看的多个文件的属性信息，如图 15-15 所示。



D4 修改日期

	A	B	C	D
1				
2			显示文件列表	
3				
4	文件名称	文件大小	文件类型	修改日期
5	E:\常用软件\GooglePinyinInstaller.exe	12281 KB	应用程序	2011-8-15 9:08
6	E:\常用软件\easyMule-Beta-Setup.exe	4395 KB	应用程序	2011-6-24 8:49
7	E:\常用软件\QQ2011.exe	43170 KB	应用程序	2011-11-29 17:31
8	E:\常用软件\pipi_459.exe	6469 KB	应用程序	2011-8-13 18:43
9	E:\常用软件\kuwosetup.exe	9638 KB	应用程序	2011-7-10 8:35
10	E:\常用软件\kmplayers.zip	11847 KB	WinRAR ZIP 压缩文件	2011-7-2 10:10
11	E:\常用软件\iTunesSetup2.exe	78978 KB	应用程序	2011-4-27 20:42
12	E:\常用软件\Install_WLMessenger.exe	9594 KB	应用程序	2011-7-13 16:45
13				
14				

图 15-15 显示所查看的多个文件的属性信息

从结果可以看出，使用 FileDialog 对象返回的文件名称包含文件所在的路径，这是与使用 VBA 函数功能所不同的地方。



15.2 将文本导入工作表

使用 VBA 读写文本文件的基本步骤是，首先使用 Open 语句打开文件，然后使用 Seek 函数指定文件中的位置，接着从文件中读取数据或将数据写入文件，最后使用 Close 语句关闭文件。

将数据导入到工作表中可分为导入某个特定的单元格和导入某个单元格区域两种情况。

15.2.1 将文本导入单元格

将文本文件中的内容导入一个单元格，具体操作步骤如下。

01 首先编写一个文本数据表，并保存在 D 盘的 Employee 文件夹中，将其命名为“员工信息.txt”，其内容如图 15-16 所示。



图 15-16　文本数据表

02 新建一个空白工作簿，并将其保存为“将文本导入工作表”，然后将 Sheet1 工作表保存为“导入员工信息表”，如图 15-17 所示。

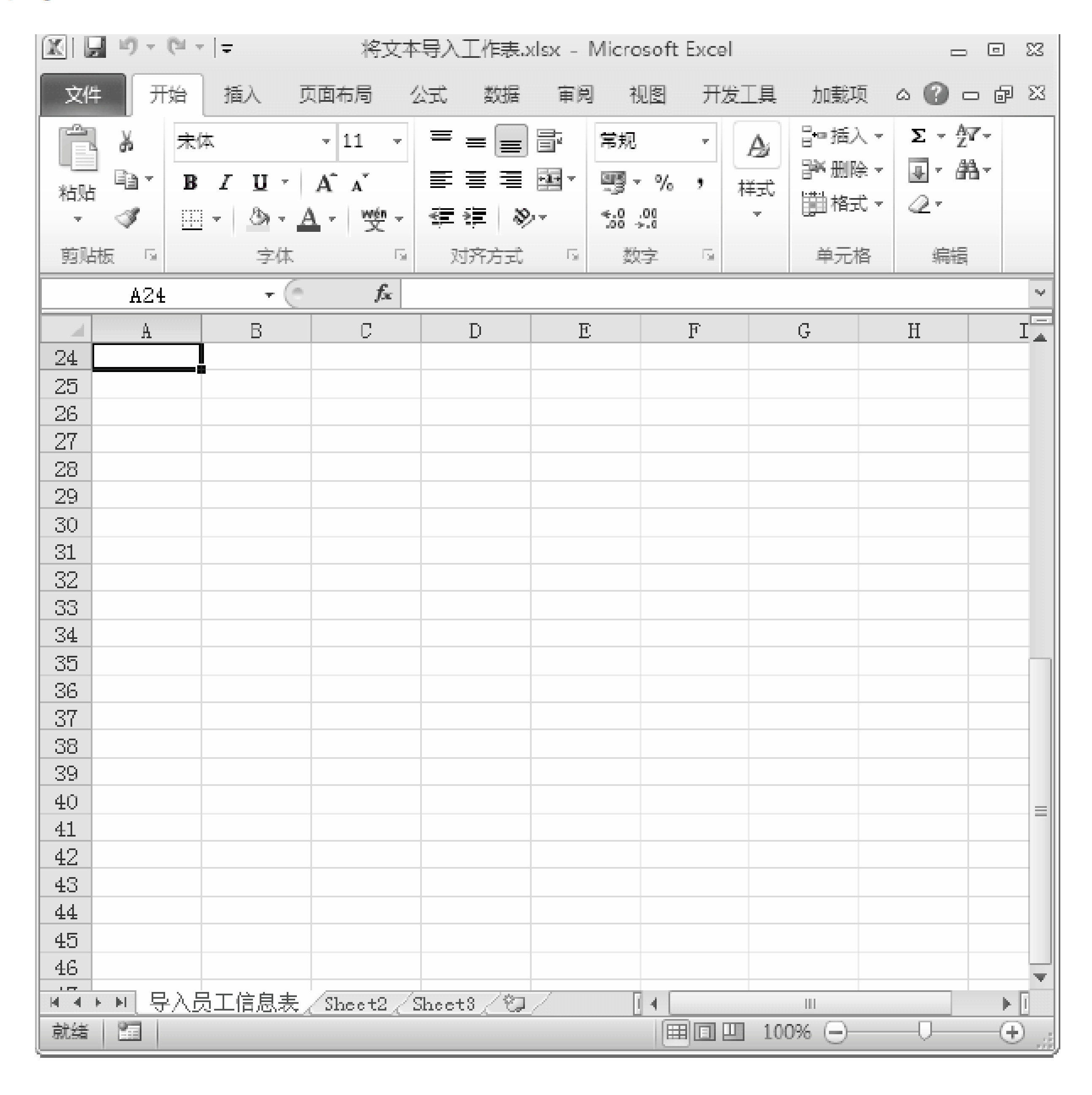


图 15-17　新建工作簿

03 在"导入员工信息表"工作表中，切换到【开发工具】选项卡，单击【控件】选项组中的 Visual Basic 按钮，打开 VBA 代码编辑窗口。双击进入"导入员工信息表"工作表的代码编辑窗口，输入下面的程序代码，如图 15-18 所示。

```
Private Sub Worksheet_Activate()
Dim i As Long
Dim Data As Variant
Open " D:\Employee\员工信息.txt " For Input As #1
i = 0
Do Until EOF(1)
    Line Input #1, Data   '读取完整的一行数据
    ActiveCell.Offset(i, 0) = Data
    i = i + 1
Loop
Close #1
End Sub
```

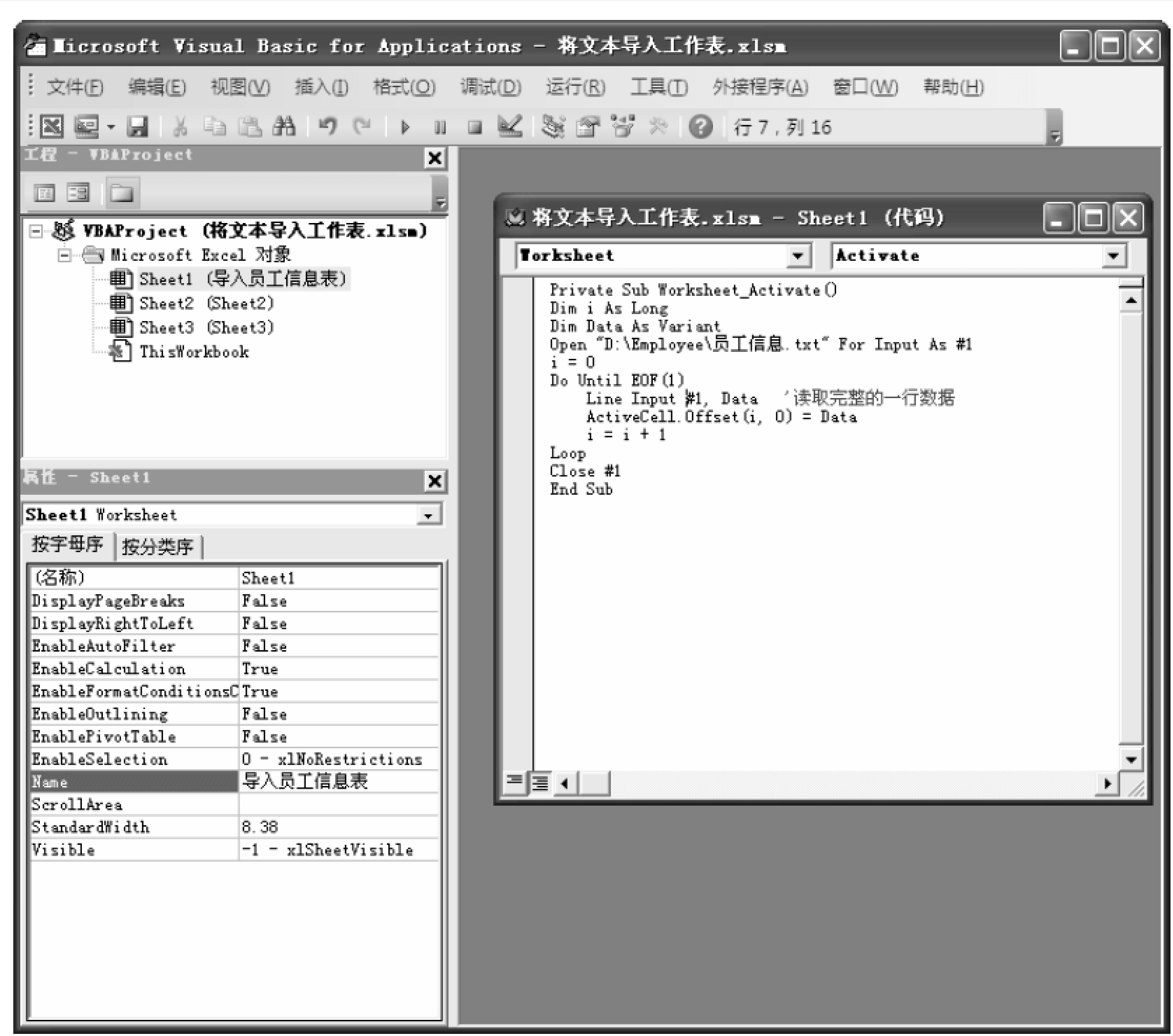


图 15-18 输入代码

04 在这段代码中首先使用 Open 方法按照路径"D:\Employee\员工信息.txt"找到"员工信息.txt"文本文件，并指定其编号为 1。这里的文件编号的指定是任意的，用户可以将文件编号指定为其他数值，只要在使用文件编号引用文件时保持该编号的统一性即可。

在 Do Until-Loop 循环语句中使用语句"Line Input #"逐行读取该文本文件中的数据，并将每行数据输入到活动单元格开始的单元格中。最后使用 Close 方法关闭该文本文件。

【提示】如果“员工信息.txt”文件的路径和上述代码中的路径不一致，程序运行将会出错。要想快速查看“员工信息.txt”文件的路径，用户可以右击该“员工信息.txt”文本文件，在弹出的快捷菜单中选择【属性】菜单项，打开【员工信息.txt】对话框，在【常规】选项卡中的【位置】项中用户可以看到该文本文件所在位置的完整路径，如图 15-19 所示，然后使用这个完整路径替换上述代码中的数据路径即可。



图 15-19　属性对话框

在上述代码中还用到了 EOF 函数，该函数的作用是返回一个整型数值，用来表明是否已经到达打开的文件的末尾。当到达文件末尾时，返回值为 True。该函数的语法格式为：

```
EOF(filenumber)
```

其中必要的参数 filenumber 是一个包含任何有效文件号的整型数值。在读写文件的操作中使用 EOF 函数可以避免出现因试图在文件末尾进行输入而产生的错误。

05 返回“导入员工信息表”工作表中，首先选中一个单元格，这里选中 A1 单元格，然后打开 VBA 代码窗口，从中选择【运行】➤【运行子过程/用户窗体】菜单项，即可将“员工信息.txt”文本文件中的数据存放在以活动单元格 A1 开始的单元格中，如图 15-20 所示。

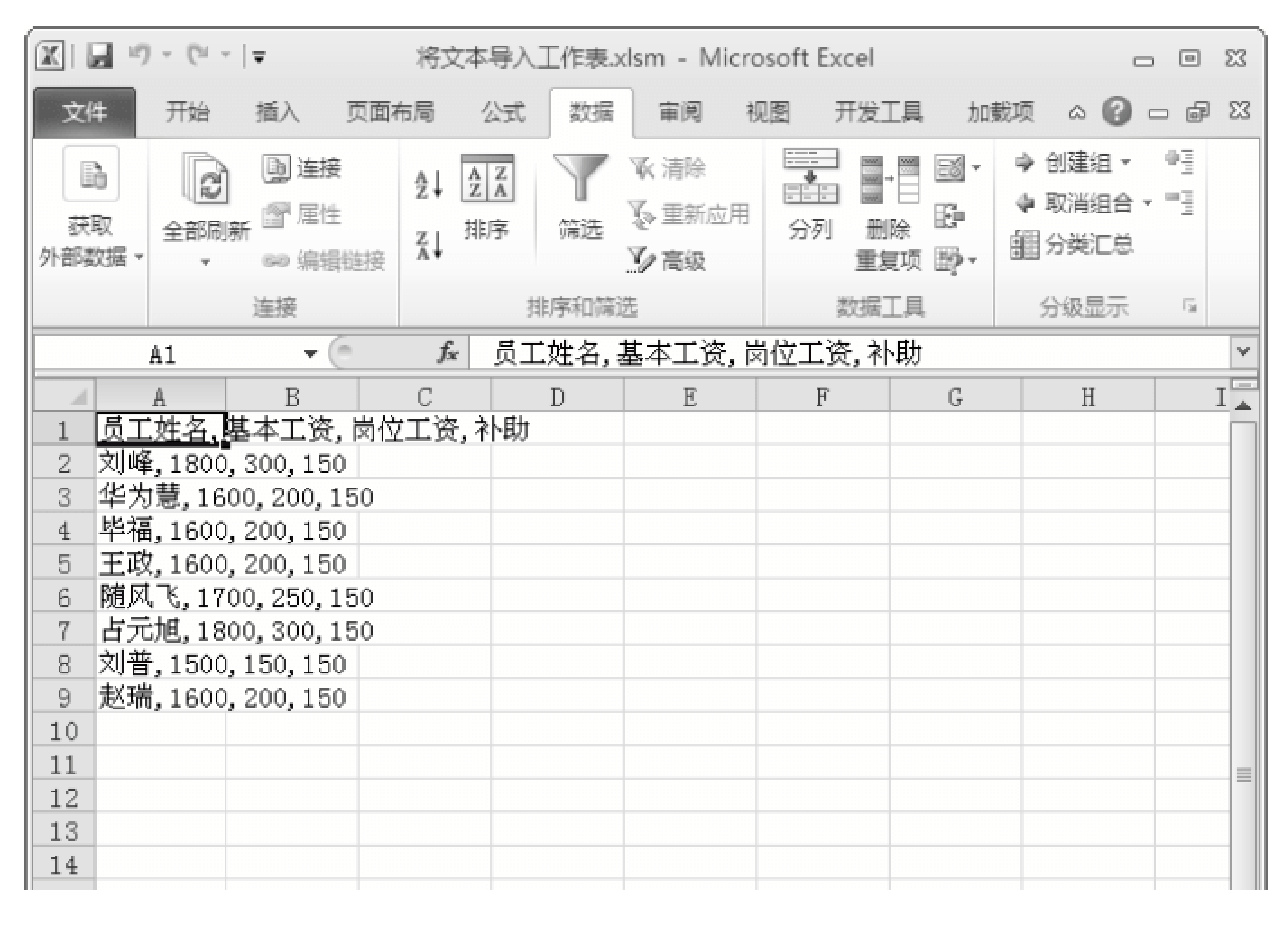


图 15-20　导入“员工信息.txt”的文本内容



上述 VBA 过程运行的结果是将“员工信息”文本文件中的每一行数据存放在同一个单元格中。一般情况下这个过程并不是非常有用。为此，用户可以使用 Excel 中的【数据】选项卡中的【分列】按钮，将数据分列在同一行的各个单元格中。

06 选中 A1:A9 单元格区域，如图 15-21 所示，然后切换到【数据】选项卡，单击【数据工具】选项组中的【分列】按钮，即可打开【文本分列向导-第 1 步】对话框，在【原始数据类型】选项区中选中【分隔符号】单选按钮，如图 15-22 所示。

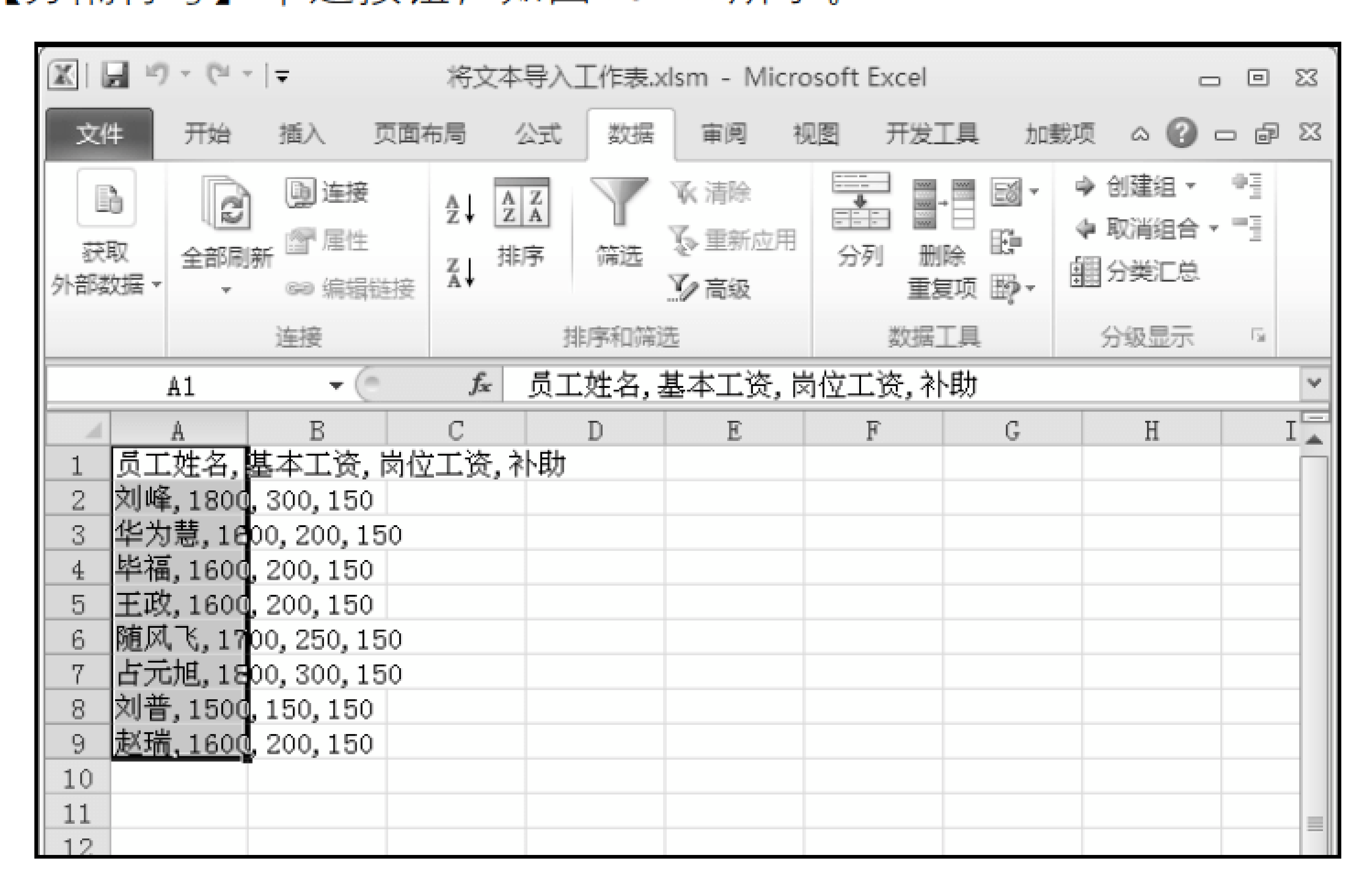


图 15-21 选中 A1:A9 单元格区域

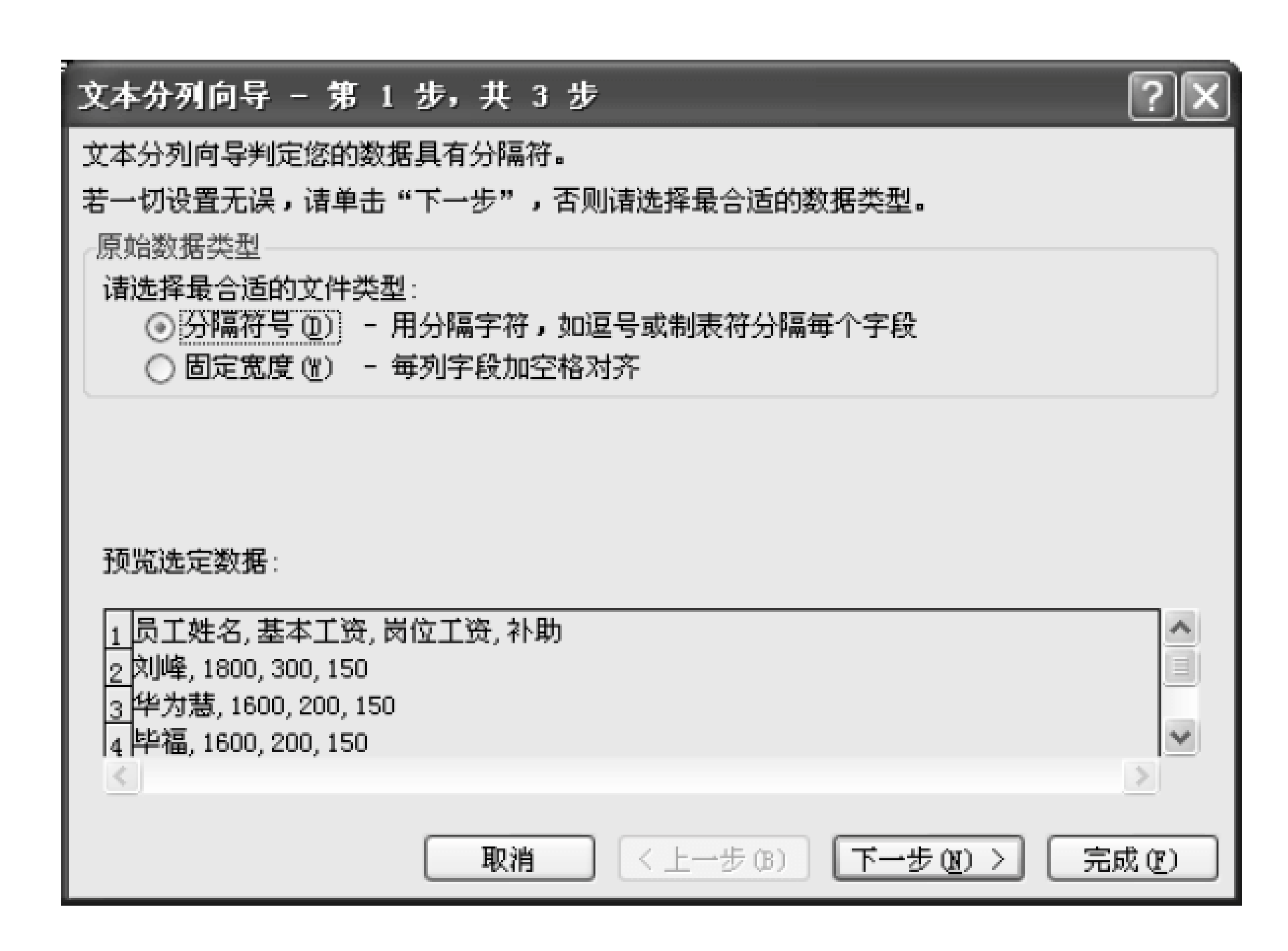


图 15-22 【文本分列向导-第 1 步】对话框

07 单击【下一步】按钮，打开【文本分列向导-第 2 步】对话框，在【分隔符号】选项区中选中【逗号】复选框，如图 15-23 所示。

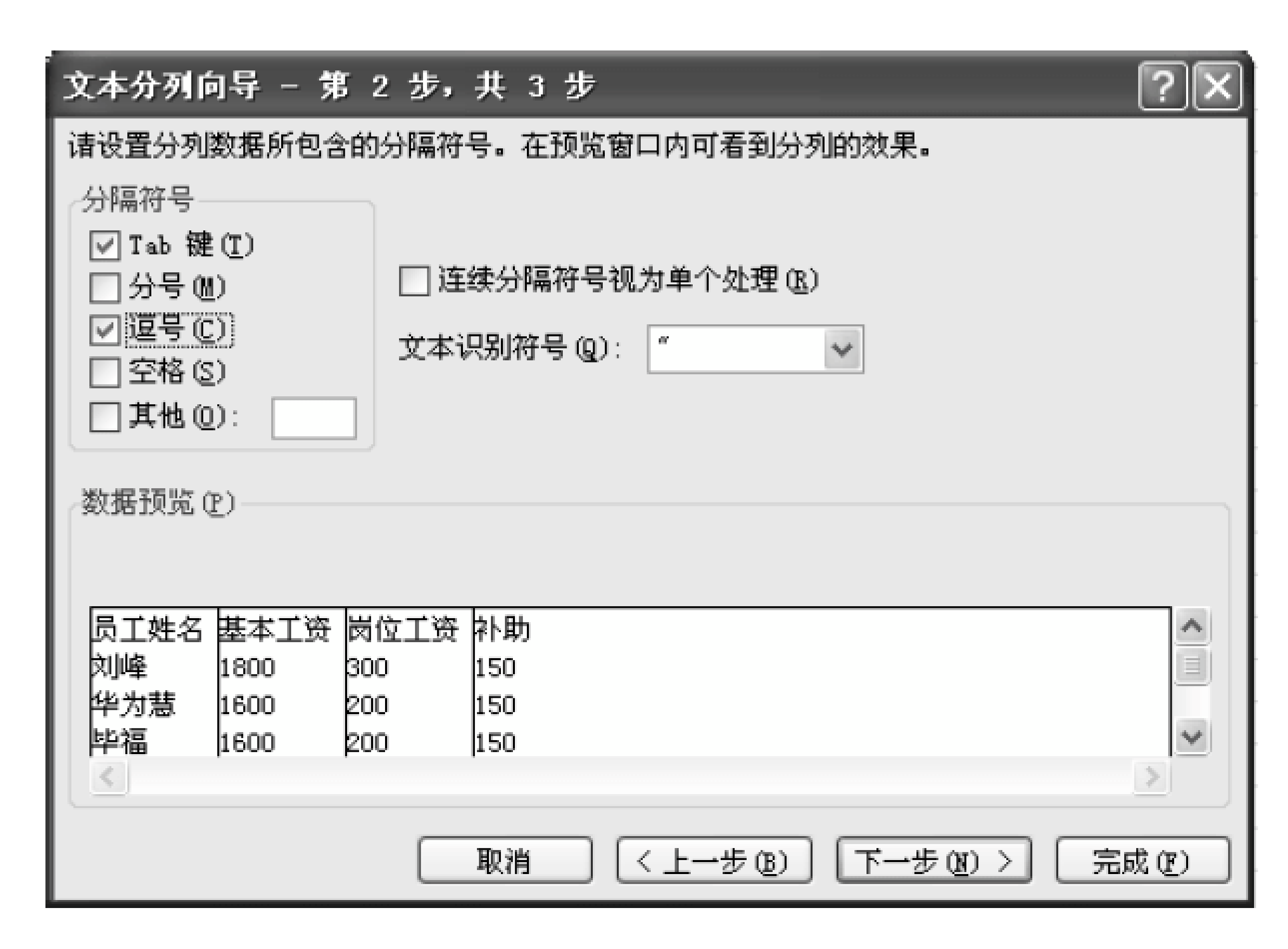


图 15-23　【文本分列向导-第 2 步】对话框

08 单击【下一步】按钮，打开【文本分列向导-第 3 步】对话框，保持该对话框中的默认设置不变，如图 15-24 所示。

09 单击【完成】按钮即可将导入的文本数据进行分列，如图 15-25 所示。



图 15-24　【文本分列向导-第 3 步】对话框



	A	B	C	D	E	F
1	员工姓名	基本工资	岗位工资	补助		
2	刘峰	1800	300	150		
3	华为慧	1600	200	150		
4	毕福	1600	200	150		
5	王政	1600	200	150		
6	随风飞	1700	250	150		
7	占元旭	1800	300	150		
8	刘普	1500	150	150		
9	赵瑞	1600	200	150		
10						
11						
12						

图 15-25　对每一行数据进行分列

15.2.2 将文本导入单元格区域

将文本导入单元格区域，具体操作步骤如下。

01 双击 Sheet2 工作表，将其重命名为“导入员工信息表 2”工作表，切换到【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中单击【ActiveX 控件】组中的【命令按钮】按钮 ，在“导入员工信息表 2”工作表中添加一个命令按钮，并将其 Caption 属性值修改为“导入文本数据”，如图 15-26 所示。

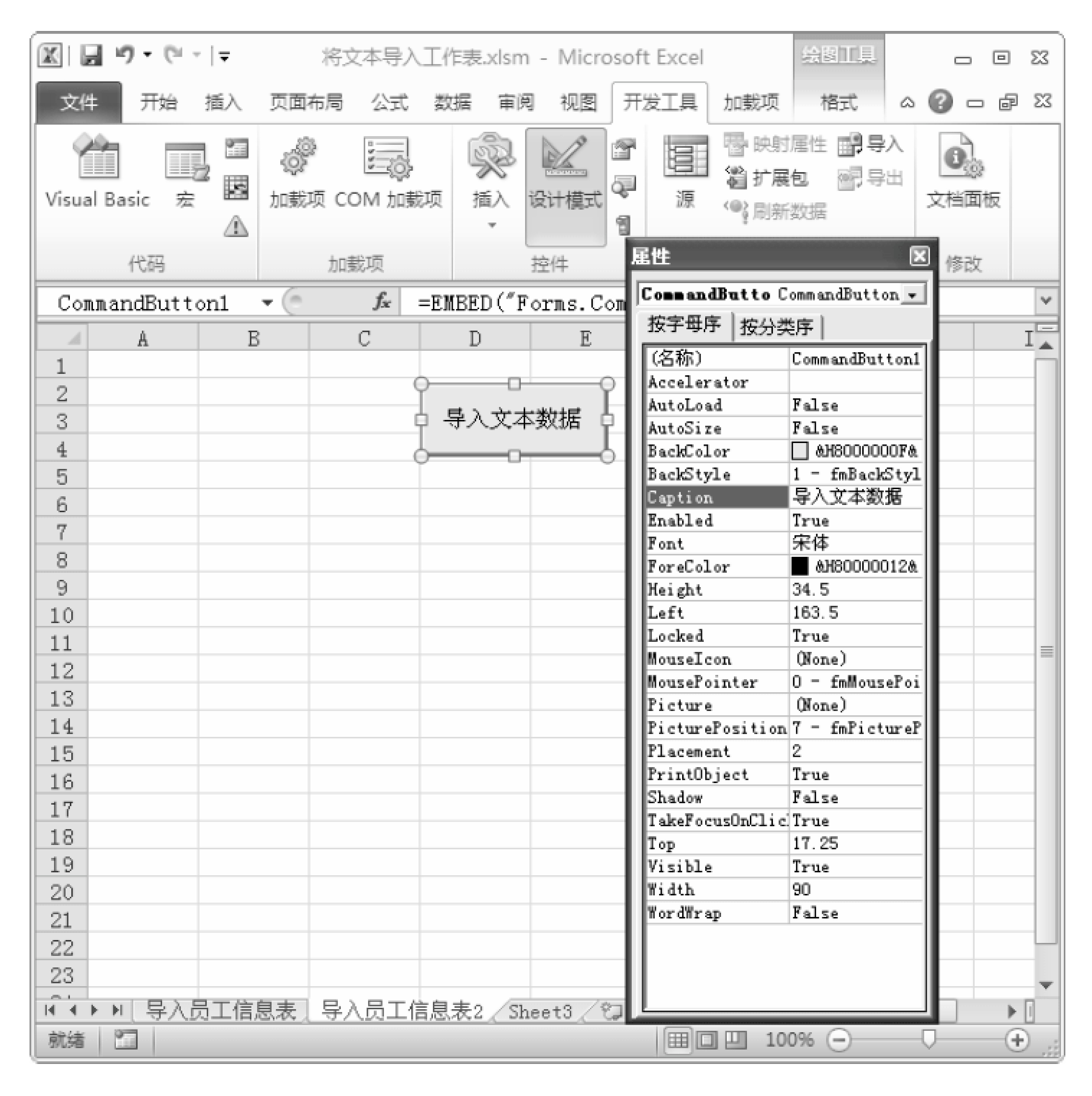


图 15-26　插入命令按钮

02 在设计模式下双击【导入文本数据】按钮，进入该按钮的代码编辑窗口，在代码窗口中添加下面的代码程序，如图 15-27 所示。

```
Private Sub CommandButton1_Click()
Dim Filename As String
Dim x As Long, y As Integer  '相对于活动单元格的偏移量
Dim str As String, temp As String
Dim Data As Variant
Dim i As Integer
On Error Resume Next  '忽略错误，执行错误语句之后的语句
Filename = " D:\Employee\员工信息.txt "
Open Filename For Input As #1
If Err <> 0 Then
    MsgBox "无法找到 " & Filename, vbCritical, "ERROR"
    Exit Sub
End If
x = 0
y = 0
str = ""
Application.ScreenUpdating = False    '忽略屏幕刷新
Do Until EOF(1)
    Line Input #1, Data
    For i = 1 To Len(Data)
        temp = Mid(Data, i, 1)  '获取 Data 中的第 i 个字符
        If temp = "," Then   '以逗号为分隔符
```

```
                ActiveCell.Offset(x, y) = str
                y = y + 1
                str = ""
            Else
                str = str & temp
                If i = Len(Data) Then  '文本文件中一行的最后一个字符
                    ActiveCell.Offset(x, y) = str
                    str = ""
                    Exit For
                End If
            End If
        Next i
        y = 0
        x = x + 1
    Loop
    Close #1  '关闭文本文件
    End Sub
```

这段代码在 Do Until-Loop 循环语句中使用了 For-Next 循环控制语句逐个字符读取文本文件中的数据，使以逗号分隔的每个数据都能各自显示在一个单元格中。需要注意的是，它不能处理包含逗号的数据。

“On Error Resume Next”的作用是当程序发生执行阶段的错误时，系统会忽略发生的错误，不修正造成错误的原因，而继续执行错误语句之后的语句。

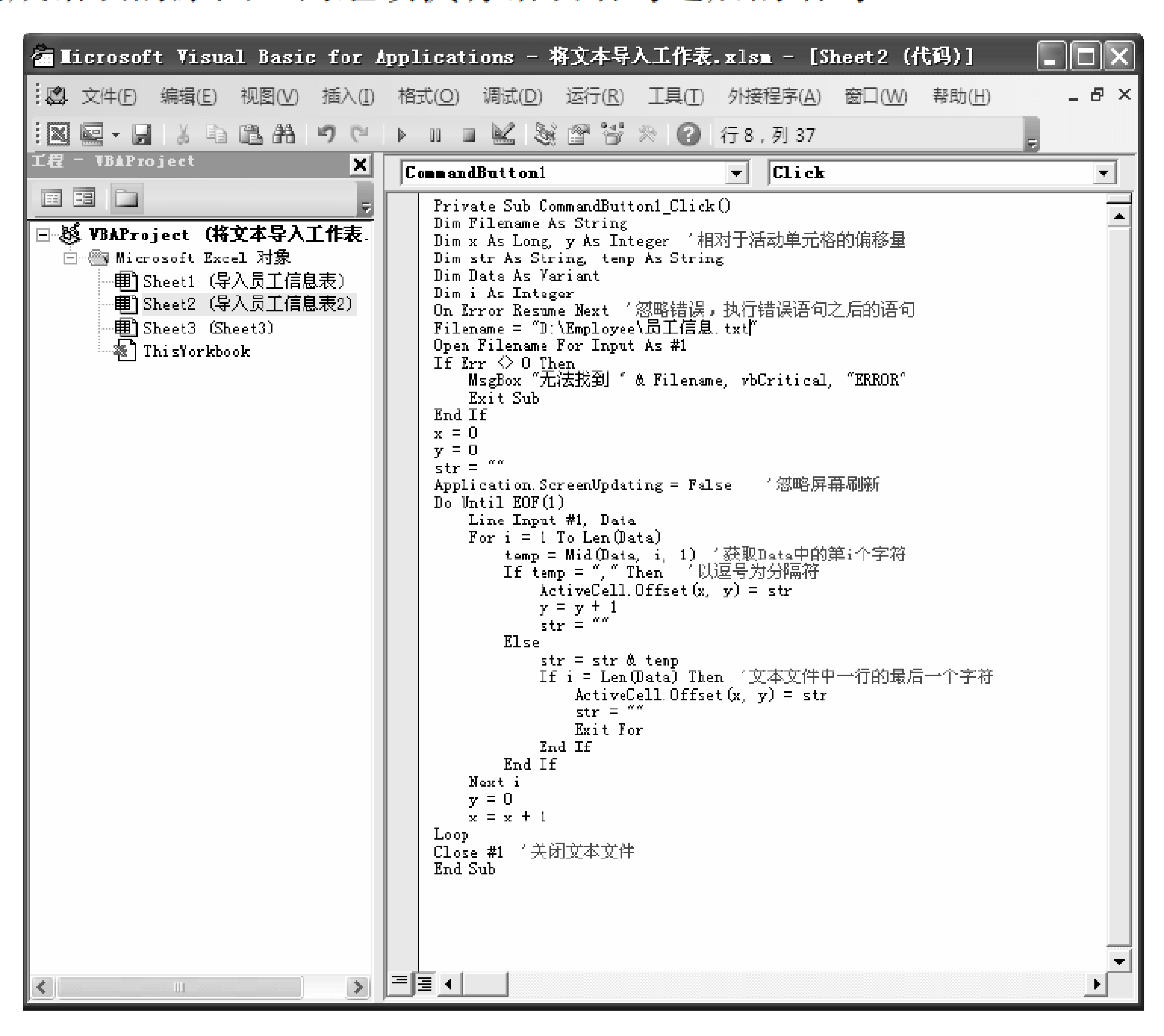


图 15-27　输入按钮响应程序

03 返回到〝导入员工信息表 2〞工作表中，单击【控件】选项组中的【退出设计模式】按钮 退出命令按钮的设计模式。任意选中一个单元格，这里选中 D10 单元格，然后单击【导入文



本数据】按钮，即可实现将数据导入到以活动单元格开头的单元格区域中，如图 15-28 所示。

	A	B	C	D	E	F	G	H
1								
2								
3				导入文本数据				
4								
5								
6								
7								
8								
9								
10				员工姓名	基本工资	岗位工资	补助	
11				刘峰	1800	300	150	
12				华为慧	1600	200	150	
13				毕福	1600	200	150	
14				王政	1600	200	150	
15				随风飞	1700	250	150	
16				占元旭	1800	300	150	
17				刘普	1500	150	150	
18				赵瑞	1600	200	150	

图 15-28　导入文本数据

【提示】导入文件的存放路径，应根据实际情况相应地调整语句：Filename =“D:\Employee\员工信息.txt”。否则，在单击【导入文本数据】按钮时会弹出如图 15-29 所示的错误提示信息。



图 15-29　错误提示信息



15.3 导出工作表中的内容

在使用 VBA 处理文件的过程中，最主要的还是处理工作表中单元格区域的内容。下面就来介绍将单元格区域中的内容分别导出到 TXT 文件和 HTML 文件的方法。

15.3.1 导出到 TXT 格式的文件

TXT 格式的文件即文本文件，可以将工作表中所选单元格区域中的内容导出到文本文件。

导出表单内容到文本文件，具体操作步骤如下：

01 在如图 15-30 所示的〝导出销售表 1〞工作表中，切换到【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中单击【ActiveX 控件】组中的【命令按钮】按钮，在〝导出销售表 1〞工作表中添加一个命令按钮，并将其 Caption 属性值修改为〝导出区域内容〞，如图 15-31 所示。

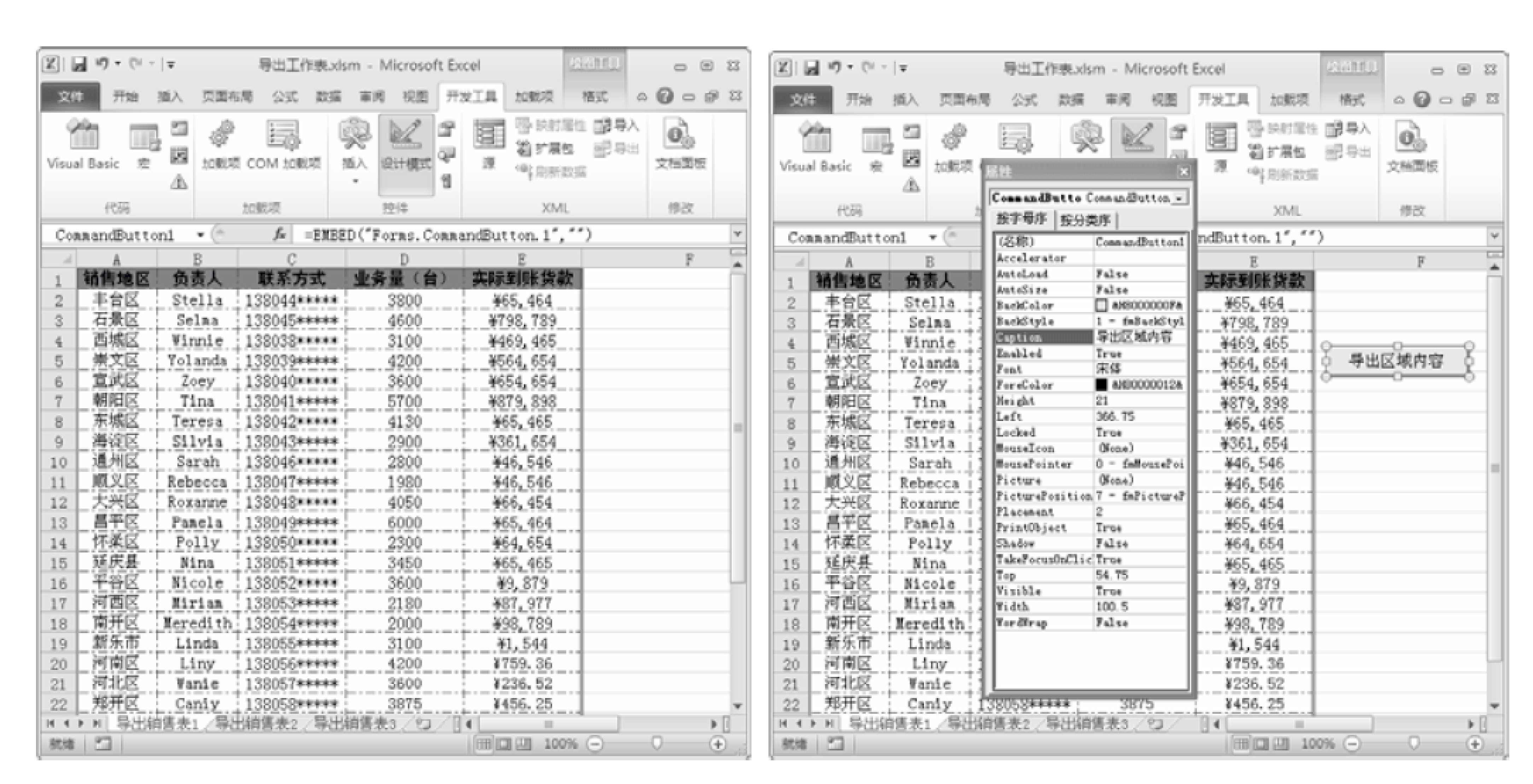


图 15-30　“导出销售表 1”工作表中的数据　　　　图 15-31　插入命令按钮

02 在设计模式下双击【导出区域内容】按钮，进入该按钮的代码编辑窗口，添加下面的代码程序，如图 15-32 所示。

```
Private Sub CommandButton1_Click()
Dim rows As Long, cols  As Integer
Dim i As Long, j As Integer
Dim tx As Variant
Dim str As String
Dim fs As Object
rows = Selection.rows.Count    '所选区域的行数
cols = Selection.Columns.Count '所选区域的列数
Set fs = CreateObject("Scripting.FileSystemObject")
'创建文本文件，若存在则覆盖原有文件
Set tx = fs.CreateTextFile("D:\Employee\导出销售数据.txt", True)
For i = 1 To rows
    str = ""
    For j = 1 To cols
        str = str & Selection.Cells(i, j) & vbTab
'每行单元格内容使用 Tab 符号连接成字符串
    Next j
    tx.WriteLine str  '写入数据，并添加一个换行符
Next i
End Sub
```



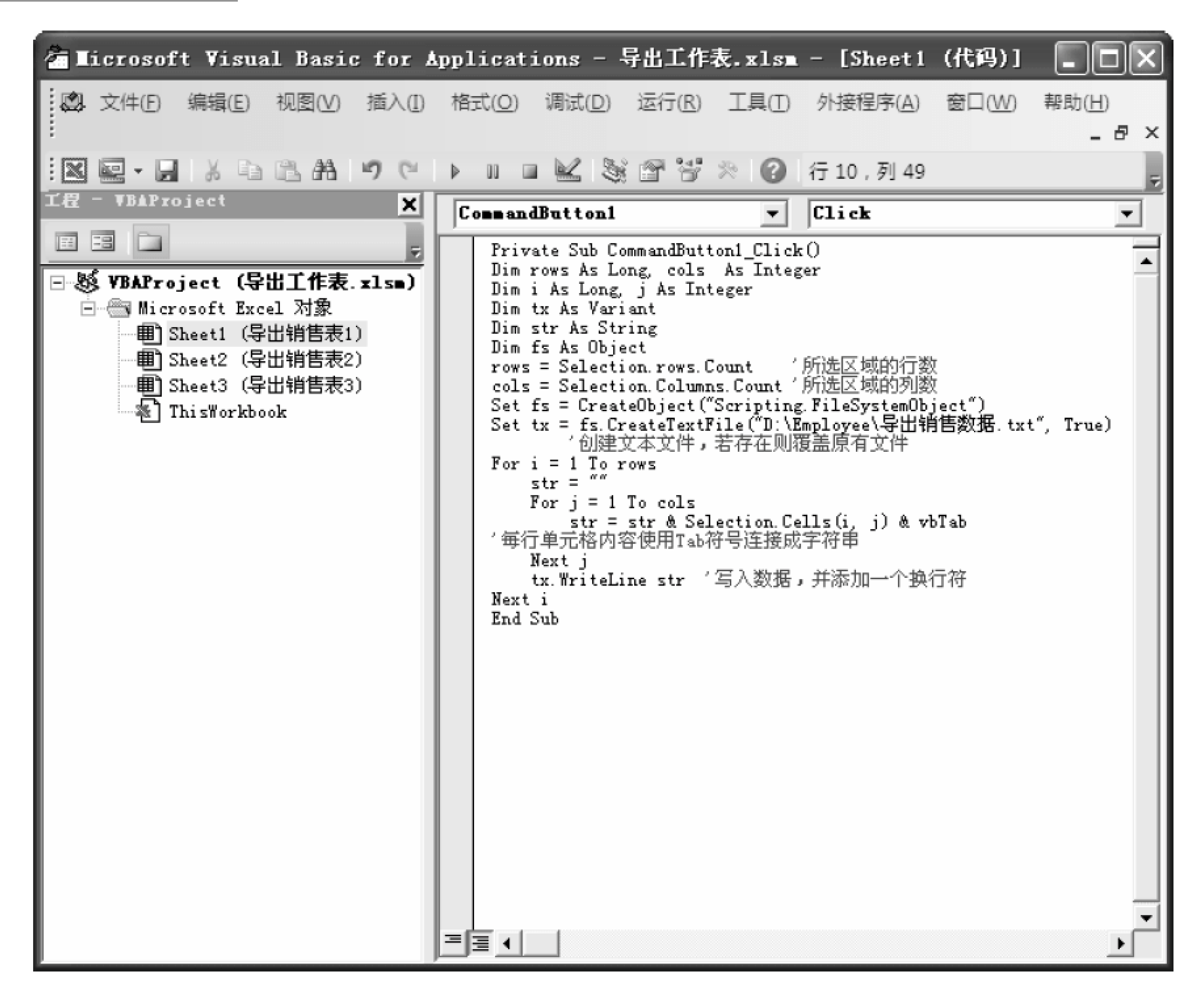


图 15-32　输入按钮响应代码

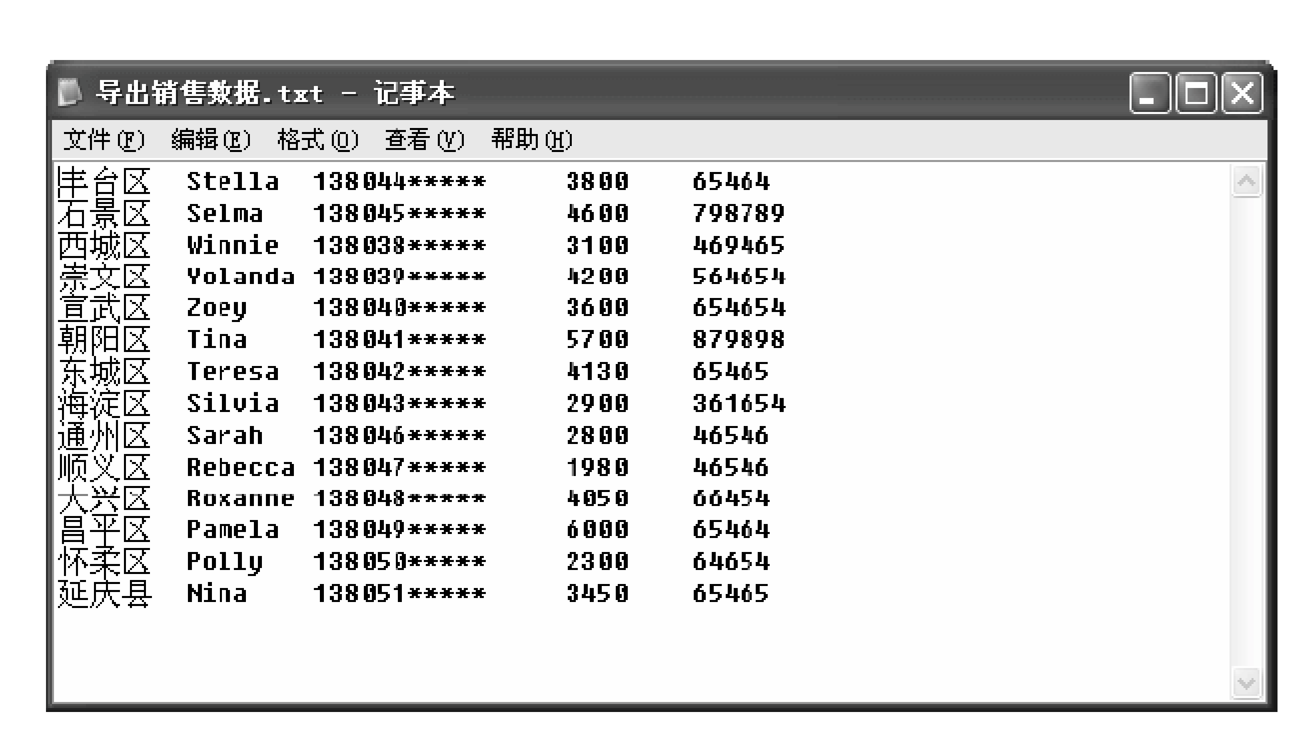
这段代码的作用是首先获得用户所选单元格区域的行数和列数，然后使用 Create TextFile 方法创建文本文件。Create TextFile 方法的语法格式为：

```
Object.Create TextFile(filename[,overwrite[,unicode]])
```

参数 filename 为必需项，指明要创建的文件的完整路径。参数 overwrite 为可选项，布尔类型，指明是否覆盖原有文件，为 True 时覆盖，为 False 时不覆盖。参数 Unicode 为可选项，布尔类型，指明文件是以何种方式创建，为 True 时以 Unicode 方式创建，为 False 时以 ASCII 方式创建。

代码中使用的是 WriteLine 方法写入数据，但会自动在写入的数据之后添加一个换行符。如果写入的数据为空，只输入一个换行符即可。

01 返回“导出销售表 1”工作表中，单击【控件】选项组中的【退出设计模式】按钮退出命令按钮的设计模式。选中一个单元格区域，这里选择 A2:E15 单元格区域，然后单击【导出所选区域内容】按钮，系统则自动在指定位置创建一个名为“导出销售数据.txt”的文本文件，打开该文件，内容显示如图 15-33 所示。



导出销售数据.txt - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

丰台区	Stella	138044*****	3800	65464
石景区	Selma	138045*****	4600	798789
西城区	Winnie	138038*****	3100	469465
崇文区	Yolanda	138039*****	4200	564654
宣武区	Zoey	138040*****	3600	654654
朝阳区	Tina	138041*****	5700	879898
东城区	Teresa	138042*****	4130	65465
海淀区	Silvia	138043*****	2900	361654
通州区	Sarah	138046*****	2800	46546
顺义区	Rebecca	138047*****	1980	46546
大兴区	Roxanne	138048*****	4050	66454
昌平区	Pamela	138049*****	6000	65464
怀柔区	Polly	138050*****	2300	64654
延庆县	Nina	138051*****	3450	65465

图 15-33　导出的内容

15.3.2　导出到 HTML 格式的文件

HTML 格式的文件仅仅是包含特殊的格式化标签的文本文件，这些格式化的标签通常用来描述数据信息是如何显示在浏览器中的。将单元格区域中的内容导出到 HTML 格式的文件就是将单元格区域的内容保存为网页格式。在 Excel 中可以使用菜单命令和 VBA 过程两种方法将单元格区域的内容保存为网页格式。

1. 使用 Excel 菜单命令

使用 Excel 菜单命令将所选单元格区域中的内容保存为网页文件，具体操作步骤如下。

01 打开要导出的数据文件，如图 15-34 所示。在"导出销售表 2"工作表中选择 B2:F23 单元格区域，然后选择【文件】➤【另存为】菜单命令，即可打开【另存为】对话框。

	A	B	C	D	E	F	G
1		销售地区	负责人	联系方式	业务量（台）	实际到账货款	
2		西城区	Winnie	138038*****	3100	¥469,465	
3		崇文区	Yolanda	138039*****	4200	¥564,654	
4		宣武区	Zoey	138040*****	3600	¥654,654	
5		朝阳区	Tina	138041*****	5700	¥879,898	
6		东城区	Teresa	138042*****	4130	¥65,465	
7		海淀区	Silvia	138043*****	2900	¥361,654	
8		丰台区	Stella	138044*****	3800	¥65,464	
9		石景区	Selma	138045*****	4600	¥798,789	
10		通州区	Sarah	138046*****	2800	¥46,546	
11		顺义区	Rebecca	138047*****	1980	¥46,546	
12		大兴区	Roxanne	138048*****	4050	¥66,454	
13		昌平区	Pamela	138049*****	6000	¥65,464	
14		怀柔区	Polly	138050*****	2300	¥64,654	
15		延庆县	Nina	138051*****	3450	¥65,465	
16		平谷区	Nicole	138052*****	3600	¥9,879	
17		河西区	Miriam	138053*****	2180	¥87,977	
18		南开区	Meredith	138054*****	2000	¥98,789	
19		新乐市	Linda	138055*****	3100	¥1,544	
20		河南区	Liny	138056*****	4200	¥759.36	
21		河北区	Wanie	138057*****	3600	¥236.52	
22		郑开区	Caniy	138058*****	3875	¥456.25	
23		黄葛区	Lovey	138059*****	4892	¥156.20	
24							
25							

导出销售表1　导出销售表2　导出销售表3

图 15-34　要导出的 Excel 文件

02 从中选择保存文件的位置，在【文件名】文本框中输入"导出工作表.htm"，在【保存

类型】下拉列表中选择“网页”，并在【保存】选项组中选择【选择（E）:$B$2:$F$23】单选按钮，如图 15-35 所示，设置完之后，单击【保存】按钮，即可完成 HTML 文件的导出。

03 设置完毕后，如果需要修改发布的内容和形式，可以单击【发布】按钮，打开【发布为网页】对话框，从中可以修改发布内容和发布形式。例如，可以修改选择的区域、发布文件的位置和名称（这里修改发布文件的名称为 page. mht）等，如图 15-36 所示。



图 15-35 【另存为】对话框

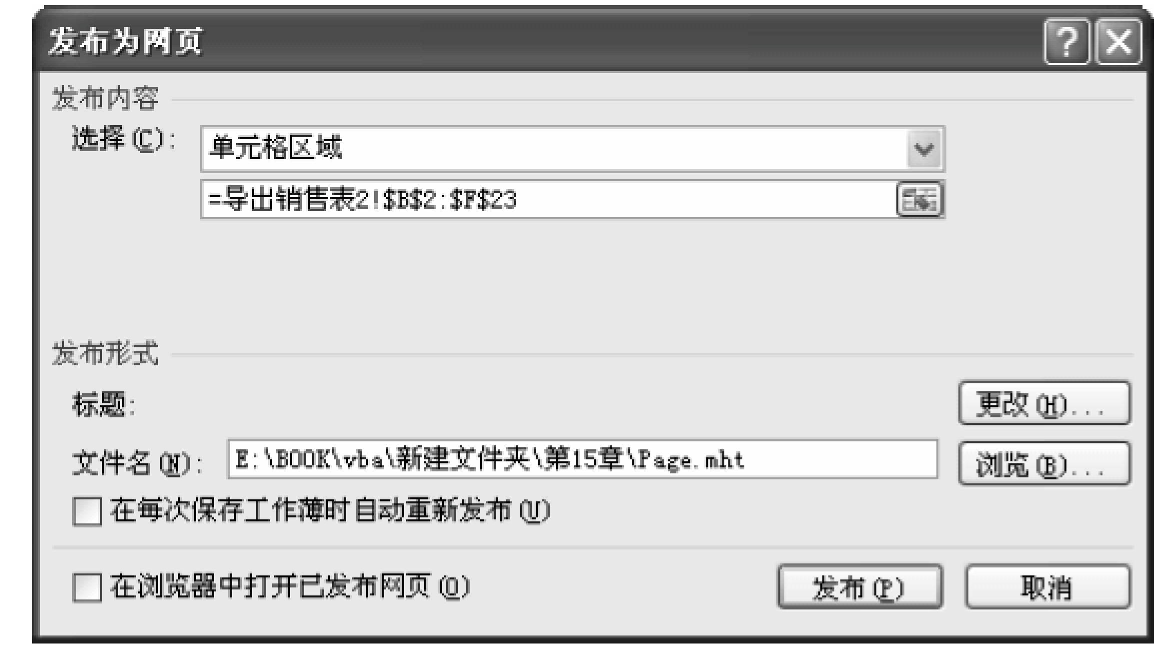
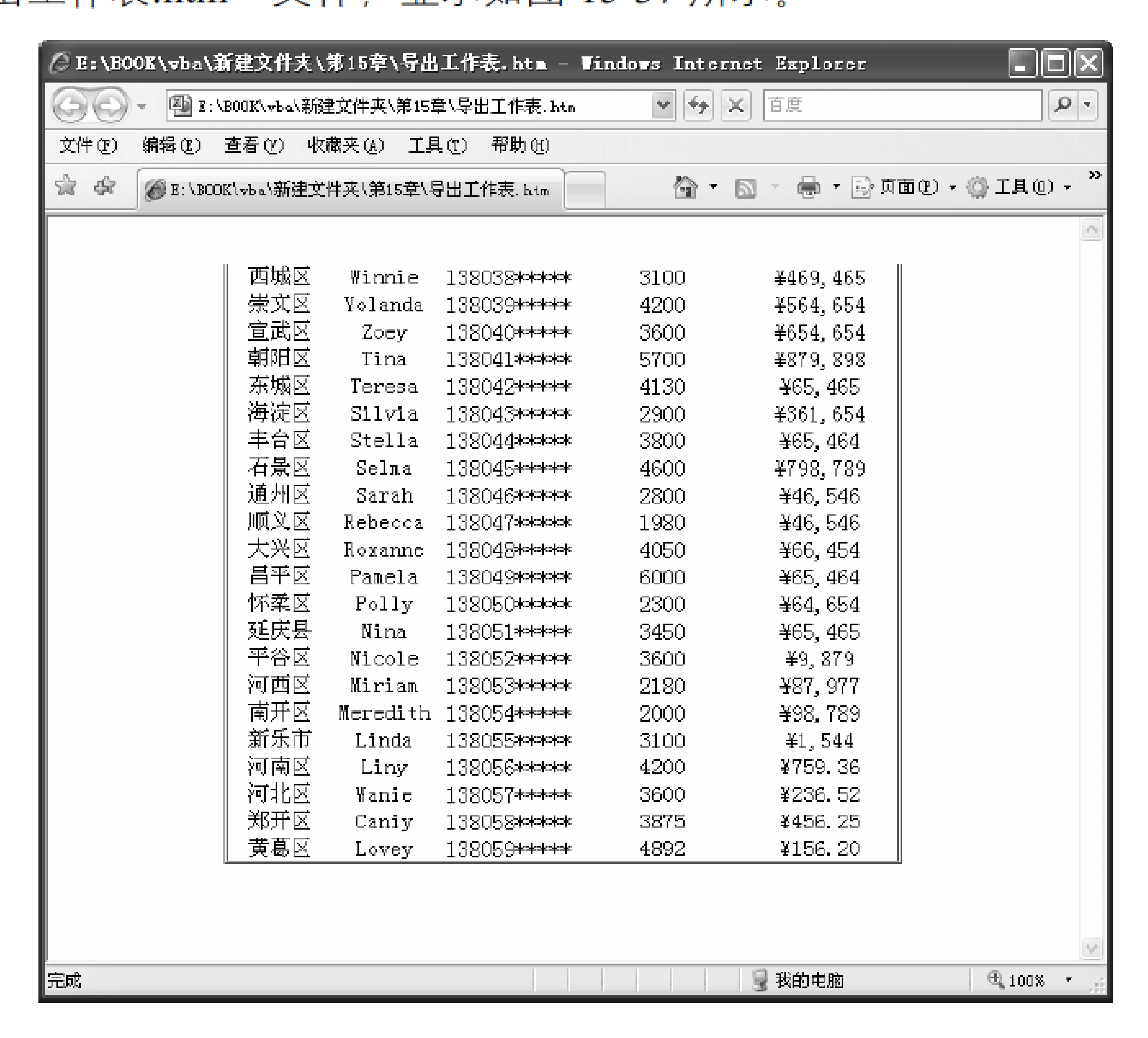


图 15-36 【发布为网页】对话框

04 修改完毕后，单击【发布】按钮，即可将选择的单元格区域内容保存为 HTML 格式的网页。打开“导出工作表.htm”文件，显示如图 15-37 所示。



西城区	Winnie	138038*****	3100	¥469, 465
崇文区	Yolanda	138039*****	4200	¥564, 654
宣武区	Zoey	138040*****	3600	¥654, 654
朝阳区	Tina	138041*****	5700	¥879, 898
东城区	Teresa	138042*****	4130	¥65, 465
海淀区	Silvia	138043*****	2900	¥361, 654
丰台区	Stella	138044*****	3800	¥65, 464
石景区	Selma	138045*****	4600	¥798, 789
通州区	Sarah	138046*****	2800	¥46, 546
顺义区	Rebecca	138047*****	1980	¥46, 546
大兴区	Roxanne	138048*****	4050	¥66, 454
昌平区	Pamela	138049*****	6000	¥65, 464
怀柔区	Polly	138050*****	2300	¥64, 654
延庆县	Nina	138051*****	3450	¥65, 465
平谷区	Nicole	138052*****	3600	¥9, 879
河西区	Miriam	138053*****	2180	¥87, 977
南开区	Meredith	138054*****	2000	¥98, 789
新乐市	Linda	138055*****	3100	¥1, 544
河南区	Liny	138056*****	4200	¥759. 36
河北区	Wanie	138057*****	3600	¥236. 52
郑开区	Caniy	138058*****	3875	¥456. 25
黄岛区	Lovey	138059*****	4892	¥156. 20

图 15-37 导出的 HTML 文件效果

2. 使用 VBA 过程

使用 VBA 过程将所选单元格导出到 HTML 文件中的方法有一个独特的优点，即不产生冗长的 HTML 代码。但是这种方法也有不足之处，那就是不能处理合并的单元格、不能维护原有单元格格式。

使用 VBA 过程导出 HTML 文件的，具体操作步骤如下。

01 在打开的“导出销售表 3”工作表中，切换到【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中单击【ActiveX 控件】组中的【命令按钮】按钮 ，在“导出销售表 3”工作表中添加一个命令按钮，并将其 Caption 属性值修改为“导出为网页”，如图 15-38 所示。

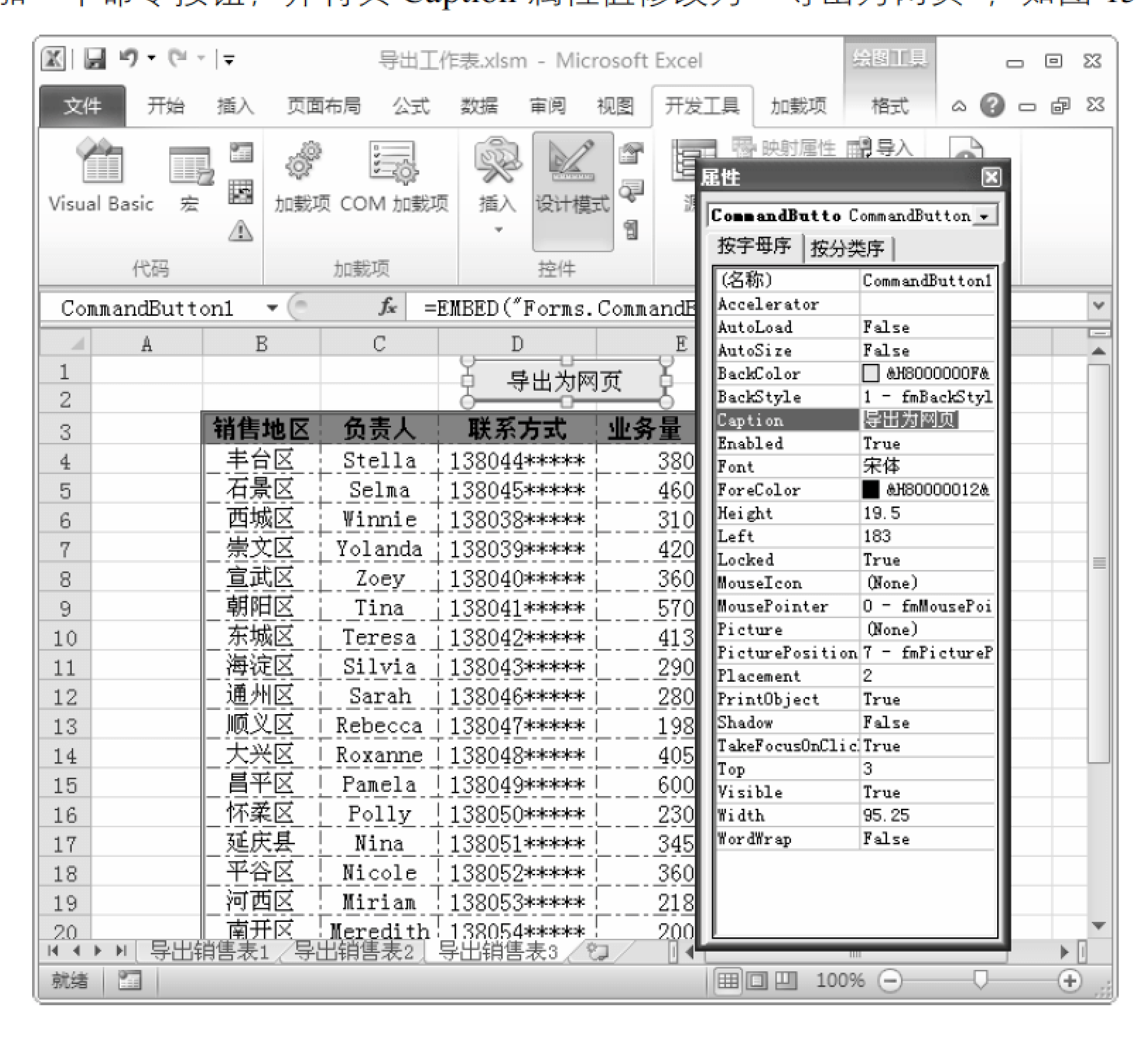


图 15-38　插入命令按钮

02 在设计模式下双击【导出为网页】按钮，进入该按钮的代码编辑窗口，添加下面的程序代码，如图 15-39 所示。

```
    Private Sub CommandButton1 Click()
    Dim Filename As Variant
    Dim temp1 As String, temp2 As String
    Dim str As String
    Dim i As Long, j As Integer
    Filename    =    Application.GetSaveAsFilename(InitialFileName:="HTML.htm",
fileFilter:="HTML Files(*.htm), *.htm")
    If Filename = False Then Exit Sub
    'Open Filename For Output As #1
    Open Filename For Append As #1
    Print #1, "<TABLE BORDER=1 CELLPADDING=3>"   '生成 HTML 标签
    For i = 1 To Selection.rows.Count
        Print #1, "<TR>"
        For j = 1 To Selection.Columns.Count  '单元格中的内容居右
            temp1 = "<TD ALIGN=RIGHT>"
            temp2 = "</TD>"
```




```
            If Selection.Cells(i, j).Font.Bold Then  '若原字体加粗，则在网页中仍加粗
                temp1 = temp1 & "<B>"
                temp2 = "</B>" & temp2
            End If
            str = Selection.Cells(i, j)  '获取单元格内容
            Print #1, temp1 & str & temp2
        Next j
        Print #1, "</TR>"
    Next i
    Print #1, "</TABLE>"
    Close #1
    MsgBox "已将所选择的" & Selection.Count & "个单元格数据导出到 " & Filename & "文
件中！"
    End Sub
```

运行这段代码的前提是用户要确定想要导出的单元格区域，然后在程序运行中所打开的对话框中修改 HTML 文件保存的名称和路径。接下来在双重 For-Next 循环控制语句中生成相应的 HTML 标签，并写数据到这个 HTML 文件中。最后关闭这个 HTML 文件，同时系统还会自动弹出一个信息汇总的对话框。

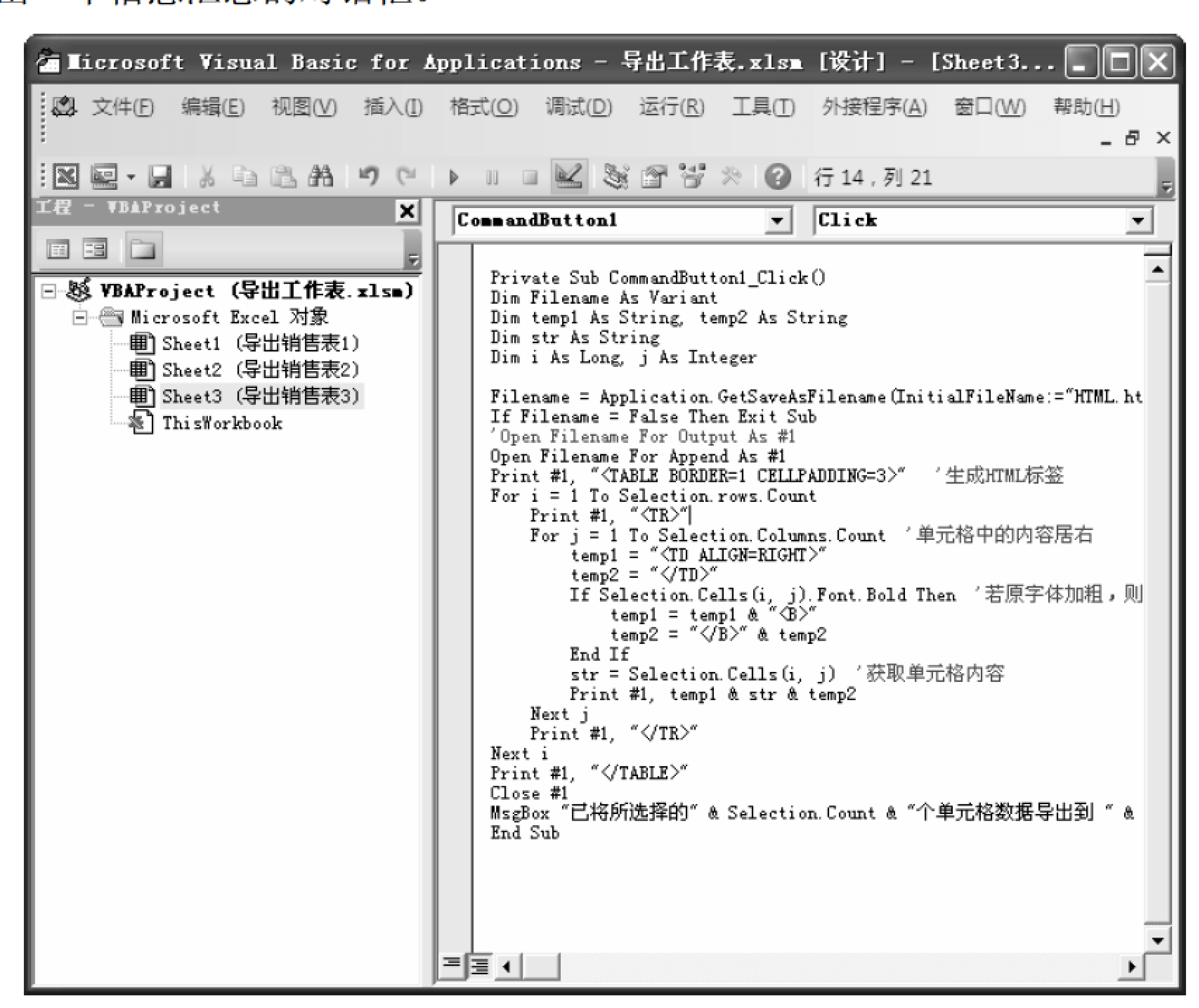


图 15-39　添加命令按钮响应程序

03 返回"导出销售表 3"工作表中，单击【控件】选项组中的【退出设计模式】按钮退出命令按钮的设计模式。选择 B3:F25 单元格区域，然后单击【导出为网页】按钮，即可打开【另存为】对话框，从中选择文件保存的位置，在【文件名】文本框中输入"VBA 导出页面文件.htm"，保存类型为 HTML Files，如图 15-40 所示。

04 设置完毕后，单击【保存】按钮，即可弹出成功导出所选内容的提示信息，如图 15-41 所示。



图 15-40　【另存为】对话框

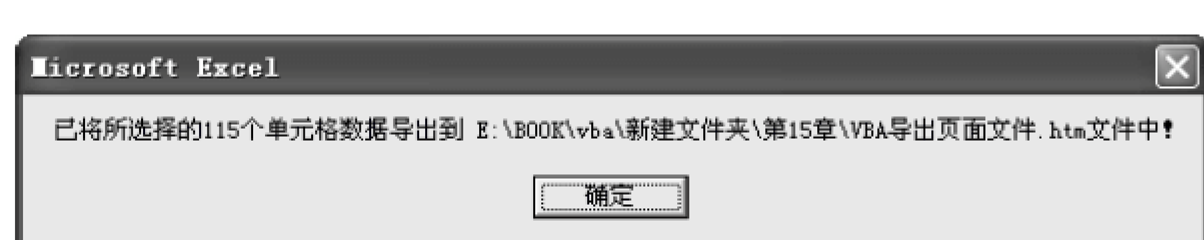


图 15-41　提示信息

05 单击【确定】按钮，即可将选中的单元格区域内容导出到指定位置。找到导出的〝VBA 导出页面文件.htm〞文件并打开，即可看到导出的文件内容，如图 15-42 所示。

【提示】如果将语句 Open Filename For Append As #1 修改为 Open Filename For Output As #1，则系统每次生成的“使用 VBA.htm”文件都是在删除原有文件的基础上重新生成的同名新文件。



15.4 记录 Excel 日志

如果需要在每次打开或关闭 Excel 时都写数据到某个文本文件中，最好的操作方法就是将能够实现该功能的 VBA 过程存放于每次启动 Excel 时打开的那个工作簿中。

在日志文件中记录 Excel 表格的打开时间，具体的操作步骤如下。

01 新建一个空白工作簿，将其另存为〝记录 Excel 日志.xlsm〞的启用宏文件。然后在〝记录 Excel 日志.xlsm〞放置的位置新建一个名为〝日志记录.txt〞的文本文件。要注意的是，这两个文件必须在同一路径下，如图 15-43 所示。



销售地区	负责人	联系方式	业务量（台）	实际到账货款
丰台区	Stella	138044*****	3800	65464
石景区	Selma	138045*****	4600	798789
西城区	Winnie	138038*****	3100	469465
崇文区	Yolanda	138039*****	4200	564654
宣武区	Zoey	138040*****	3600	654654
朝阳区	Tina	138041*****	5700	879898
东城区	Teresa	138042*****	4130	65465
海淀区	Silvia	138043*****	2900	361654
通州区	Sarah	138046*****	2800	46546
顺义区	Rebecca	138047*****	1980	46546
大兴区	Roxanne	138048*****	4050	66454
昌平区	Pamela	138049*****	6000	65464
怀柔区	Polly	138050*****	2300	64654
延庆县	Nina	138051*****	3450	65465
平谷区	Nicole	138052*****	3600	9879
河西区	Miriam	138053*****	2180	87977
南开区	Meredith	138054*****	2000	98789

图 15-42　导出的文件内容



图 15-43　新建 xlsm 文件和 txt 文件

02 在"记录 Excel 日志.xlsm"工作簿中切换到【开发工具】选项卡，单击【控件】选项组中的【Visual Basic】按钮，打开 VBA 代码编辑窗口。从中双击【工程资源管理器】窗口中的 ThisWorkbook 选项，打开 ThisWorkbook 的代码窗口，从中输入记录打开"记录 Excel 日志.xlsm"工作簿时间的程序代码，如图 15-44 所示。

```
Private Sub Workbook Open()
Open ThisWorkbook.Path & "\日志记录.txt" For Append As #1
Print #1, "打开 Excel 的时间: " & Now
Close #1
End Sub
```

这段代码的作用是将打开"记录 Excel 日志.xlsm"工作簿的时间追加到"日志记录.txt"文件的末尾，追加完毕后自动关闭该文本文件。

值得注意的是，语句 Open ThisWorkbook.Path & \日志记录.txt"中的 ThisWorkbook.Path 是一个系统变量，指的是"记录 Excel 日志 .xlsm"工作簿的路径。由于在步骤 1 中已经将"记录 Excel 日志.xlsm"文件和"日志记录.txt"文件放在了同一路径下，所以这个路径也可用来表示"日志记录.txt"文件的路径。

03 保存 VBA 代码，然后关闭"记录 Excel 日志.xlsm"工作簿。重新打开该工作簿时就会发现，系统已自动将访问该工作簿的时间添加到了"日志记录.txt"文件的末尾，如图 15-45 所示。

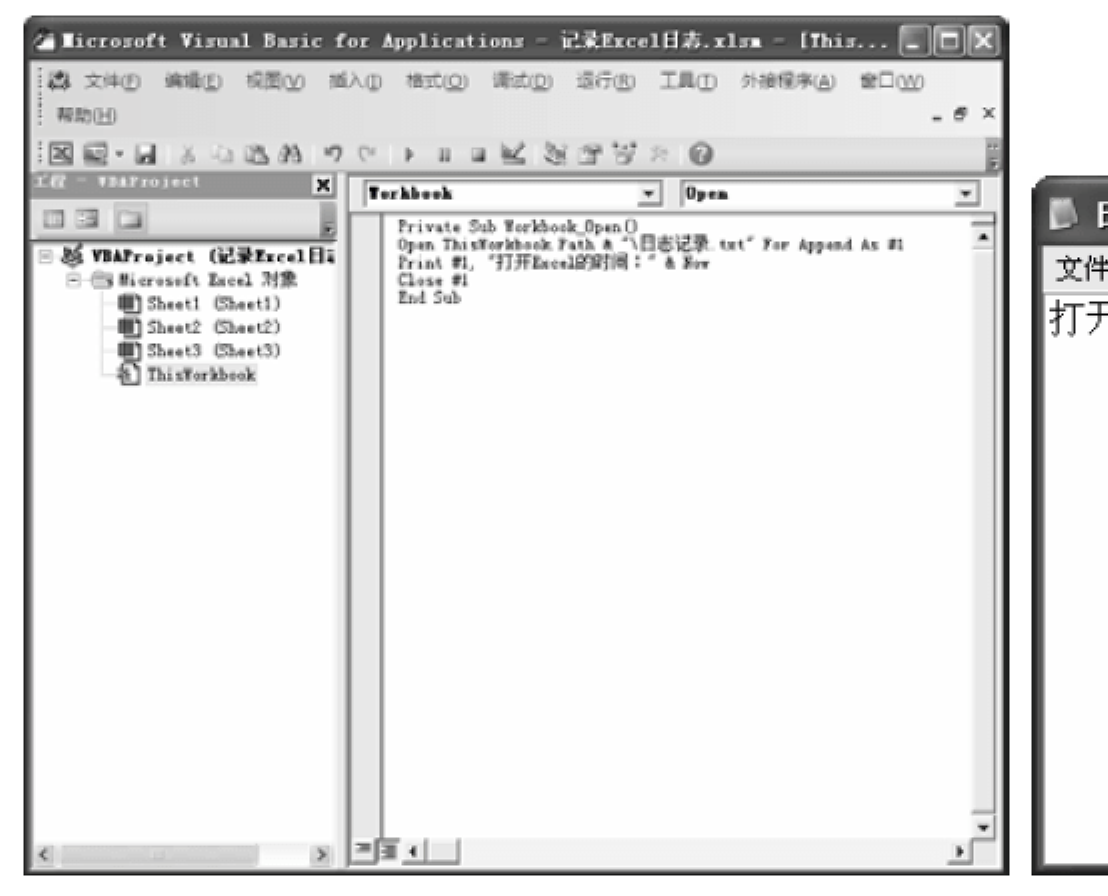


图 15-44　获取打开 Excel 的时间程序代码



图 15-45　记录打开 Excel 的时间

04 在 ThisWorkbook 代码窗口中继续输入获取关闭文件的时间代码程序，如图 15-46 所示。

```
Private Sub Workbook BeforeClose(Cancel As Boolean)
Open ThisWorkbook.Path & "\日志记录.txt" For Append As #1
Print #1, "关闭 Excel 的时间: " & Now
Close #1
End Sub
```

这段代码的作用是在每次关闭"记录 Excel 日志.xlsm"工作簿时，系统会自动将关闭时间追加到"日志记录.txt"文件的末尾，追加完毕后自动关闭该文本文件。

05 保存 VBA 代码，关闭"记录 Excel 日志.xlsm"工作簿。此时打开"日志记录.txt"文件，就会发现系统已经自动将关闭该工作簿的时间添加到了"日志记录.txt"文件的末尾，如图 15-47 所示。

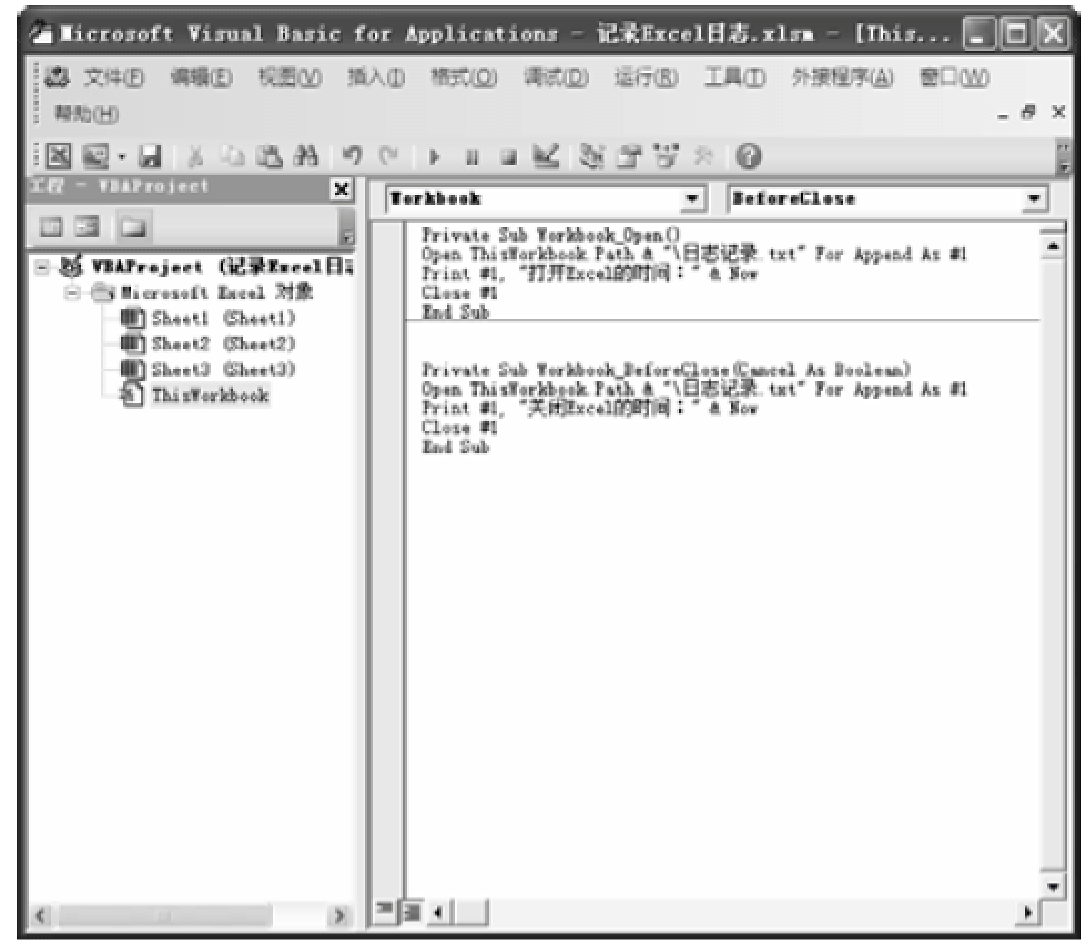


图 15-46　输入获取关闭 Excel 的时间代码程序



图 15-47　记录关闭 Excel 的时间



15.5 创建级联菜单管理工作簿

在对多个文件进行管理的时候，可以使用级联菜单的方式将多个工作表或工作簿的名字存放在菜单上，通过单击菜单上的名称就能调出相应的工作表来。使用级联菜单的方式管理多个工作簿不仅使访问文件方便快捷，而且条理清晰、便于查找。

下面就以某企业对工资数据的管理为例，介绍如何使用 Excel VBA 创建管理多个工作簿的级联菜单。

15.5.1 创建管理工资数据的工作簿

在对工资数据的管理过程中，需要统计各个月份每位员工的考勤记录和销售业绩等工资数据。通常，企业管理人员会将各个部门的工资管理数据按月存放在不同的工作簿中，即分别建立如图 15-48 所示的“三月份销售部工资管理”工作簿和如图 15-49 所示的“四月份销售部工资管理”工作簿。



图 15-48 三月份销售部工资管理



四月份销售部工资表

员工编号	姓名	基本工资	收入提成	住房补助	应扣请假费	三金
w101	刘玲	1400	400	120	60	85
w102	芮小思	1800	375	120	0	85
w103	王啸东	1400	225	120	230	85
w104	段玲华	2200	325	120	60	85
w105	郑克霜	600	425	120	200	85
w106	陈艳艳	1000	270	120	270	85
w107	贺鹏	600	280	120	800	85
w108	王英英	1400	345	120	260	85
w109	冯世雄	1400	225	120	0	85
w110	孙世宁	1400	495	120	660	85
w111	闻静	1400	225	120	400	85
w112	陈卫钶	1800	315	120	240	85
w113	陈慧军	1400	435	120	30	85
w114	李伟	1400	225	120	300	85
w115	王肖喆	2200	340	120	60	85
w116	李建刚	1400	225	120	230	85

图 15-49 四月份销售部工资管理

这里只是列举了“三月份销售部工资管理”和“四月份销售部工资管理”两个工作簿，以此为例来介绍使用 Excel VBA 进行工资数据管理的方法。

创建工资管理工作簿，具体的操作步骤如下。

01 为了实现在一个工作簿中管理多个工作簿的功能，首先需要建立一个新的工作簿，并将其保存为“销售部全年记录”，然后在该工作簿中删除 Sheet2 和 Sheet3 工作表，并双击 Sheet1 工作表标签，将其重命名为“销售部全年记录”，如图 15-50 所示。

02 在“销售部全年记录”工作表中添加说明性文字，完成管理工资数据工作簿的创建，如图 15-51 所示。

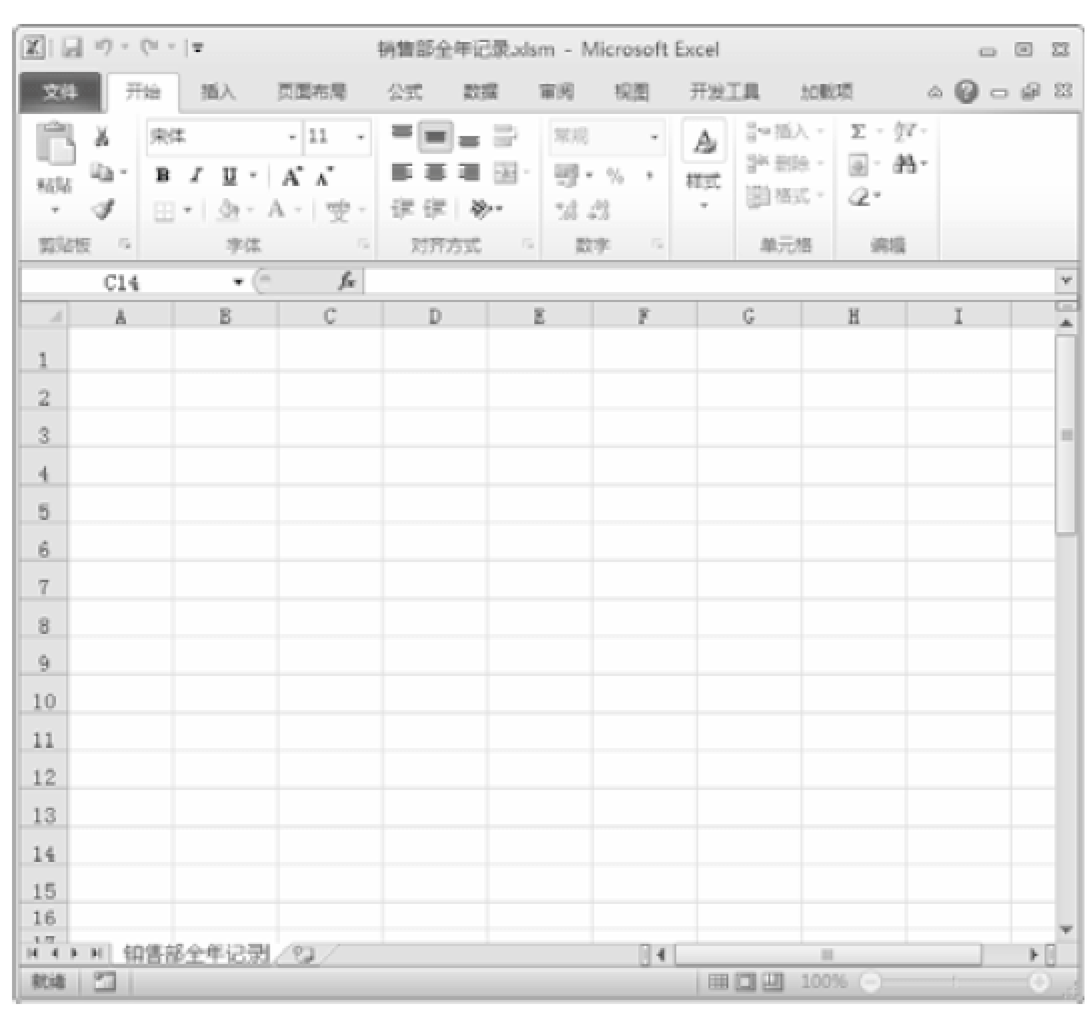


图 15-50 “销售部全年记录”工作簿

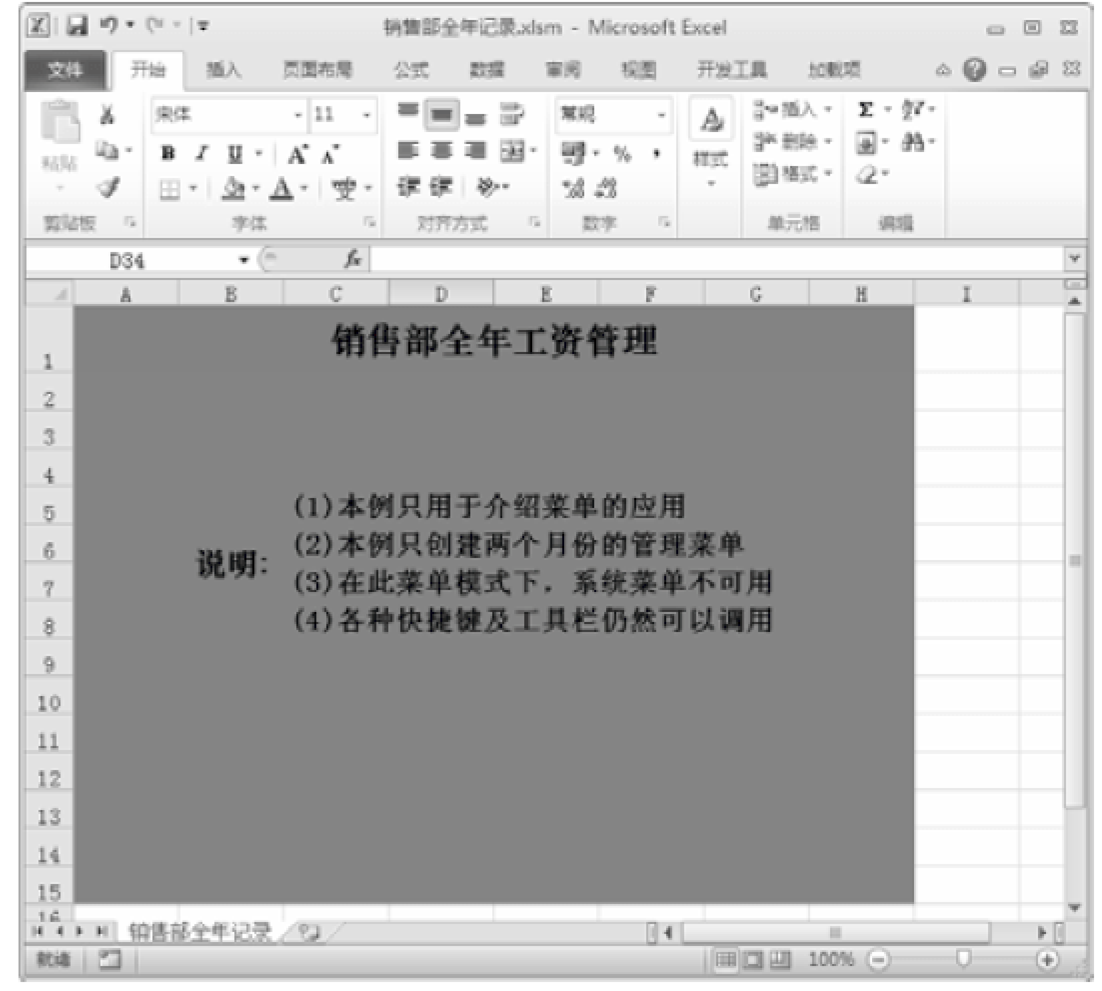


图 15-51 创建“销售部全年记录”工作表

15.5.2 编辑创建菜单的 VBA 代码

为了在“销售部全年记录”工作簿中使用级联菜单管理“三月份销售部工资管理”和“四月份销售部工资管理”两个工作簿，需要按照下面的操作步骤编辑创建菜单的 VBA 代码。

创建操作工资管理工作簿的级联菜单，操作过程如下。

01 在〝销售部全年记录〞工作表中，切换到【开发工具】选项卡，单击【控件】选项组中的 Visual Basic 按钮，打开 VBA 代码编辑窗口，从中选择【插入】➤【模块】菜单项，即可在当前工作簿中插入一个模块，然后在其【属性】窗口中的【名称】文本框中输入〝创建级联菜单〞，按下 Enter 键即可将插入的模块重命名，如图 15-52 所示。

02 在〝创建级联菜单〞模块代码窗口中添加下面的自定义创建级联菜单函数的代码，如图 15-53 所示。

```
    Sub OpenMenu() '自定义创建级联菜单的函数
        On Error Resume Next '忽略错误
        MenuBars("MyMenu").Delete '删除工作簿中原有的自定义菜单项
        MenuBars.Add ("MyMenu") '增加用户自定义的菜单项
        Sheets("销售部全年记录").Select
        MenuBars("MyMenu").Menus.Add Caption:="三月份" '增加第一个“三月份”菜单
        MenuBars("MyMenu").Menus("三月份").MenuItems.Add Caption:="三月份考勤记录表
", OnAction:="SheetOpen11"
         '在“三月份”菜单下增加“三月份考勤记录表”菜单项
        MenuBars("MyMenu").Menus("三月份").MenuItems.Add Caption:="三月份销售业绩表
", OnAction:="SheetOpen12"
         '在“三月份”菜单下增加“三月份销售业绩表”菜单项
        MenuBars("MyMenu").Menus("三月份").MenuItems.Add Caption:="三月份销售部工资
表", OnAction:="SheetOpen13"
         '在“三月份”菜单下增加“三月份销售部工资表”菜单项
        MenuBars("MyMenu").Menus("三月份").MenuItems.Add Caption:="销售部员工资料表
", OnAction:="SheetOpen14"
         '在“三月份”菜单下增加“销售部员工资料表”菜单项
        MenuBars("MyMenu").Menus.Add Caption:="四月份" '增加“四月份”菜单
        MenuBars("MyMenu").Menus("四月份").MenuItems.Add Caption:="四月份考勤记录表
", OnAction:="sheetopen21"
        MenuBars("MyMenu").Menus("四月份").MenuItems.Add Caption:="四月份销售业绩表
", OnAction:="SheetOpen22"
        MenuBars("MyMenu").Menus("四月份").MenuItems.Add Caption:="四月份销售部工资
表", OnAction:="SheetOpen23"
        MenuBars("MyMenu").Menus("四月份").MenuItems.Add Caption:="销售部员工资料表
", OnAction:="SheetOpen24"
        MenuBars("MyMenu").Activate '激活自定义菜单项
    End Sub
```

这段代码的作用是首先使用 On Error Resume Next 语句忽略错误；然后删除工作簿中原有的自定义菜单项，并重新添加用户自定义的菜单，即防止重复加载自定义菜单项；接着选择“销售部全年记录”工作表，然后在该工作表中分别添加三月份和四月份的级联菜单，并设置其响

应方式；最后使用 MenuBars（“MyMenu”）.Activate 语句来激活添加的自定义菜单项。

这段代码中主要使用 MenuBars（“MyMenu”）.Menus.Add Caption:=（“三月份”）语句来添加“三月份”的一级菜单，然后使用 MenuBars（“MyMenu”）.Menus.（“三月份”）.MenuItems.Add Caption:=“三月份考勤记录表”，OnAction:=“SheetOpen11”语句来添加二级菜单，在添加完成后自动设置其响应方式，即在用户选择某个菜单时，系统将自动调用其响应函数 SheetOpen11。通过这种方法，不仅可以创建管理数据的二级菜单，而且可以创建三级甚至四级等更多的子菜单。



图 15-52 插入“创建级联菜单”模块

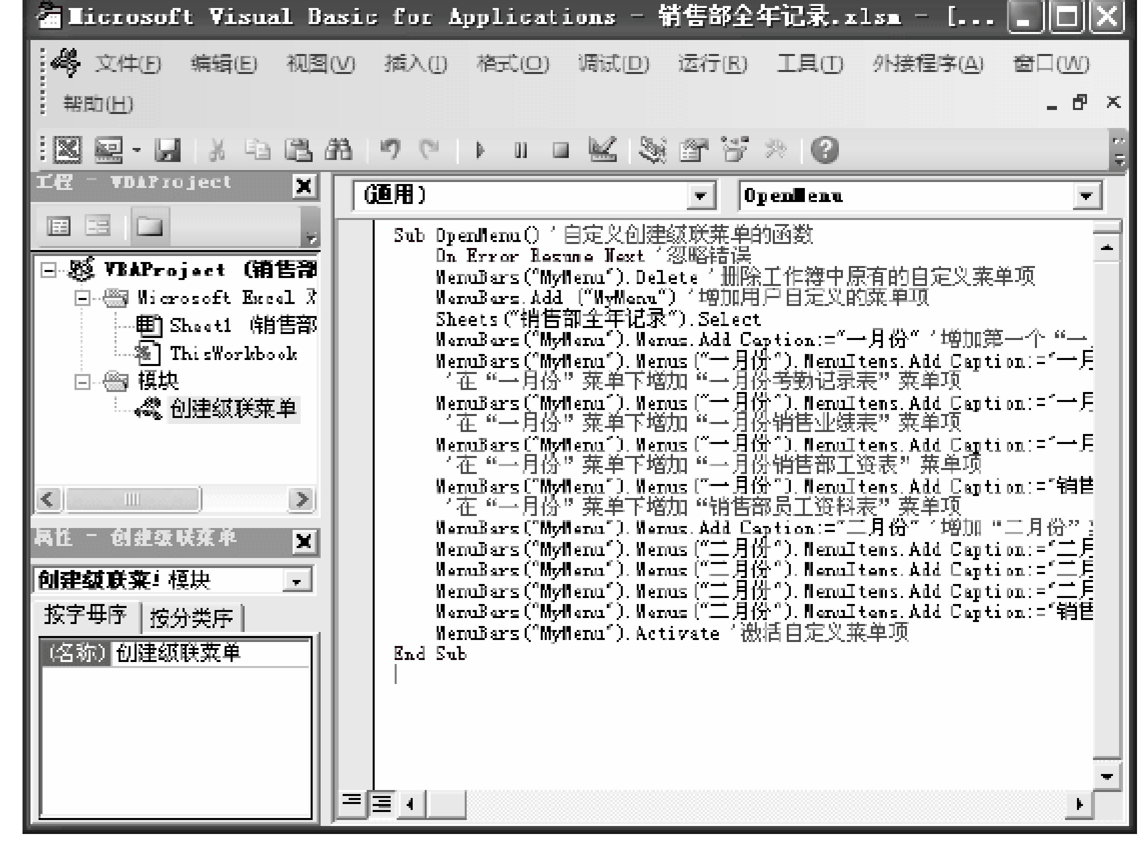


图 15-53 自定义创建级联菜单函数的代码

03 创建级联菜单时调用的响应函数都很相似，但是在菜单的响应事件中却不能传递参数，因此只能分别编辑每个菜单项的响应函数代码。于是，在上述代码窗口中添加创建级联菜单时的响应函数 SheetOpen11 代码，如图 15-54 所示。

```
Sub SheetOpen11()     '编辑创建“三月份考勤记录表”级联菜单的响应代码
Dim book As Workbook '定义一个 Workbook 变量
On Error Resume Next
Set book = Workbooks("三月份销售部工资管理.xlsm")
If book Is Nothing Then '说明“三月份销售部工资管理”工作簿没打开
   Workbooks.Open Filename:=ThisWorkbook.Path + "\三月份销售部工资管理.xlsm"
     '打开“三月份销售部工资管理”工作簿
End If
Windows("三月份销售部工资管理.xlsm").Activate
Sheets("三月份考勤记录表").Activate  '使“三月份考勤记录表”为当前活动工作表
End Sub
```

这段代码的作用是首先定义一个 Workbook 数据类型的对象变量 book，并将工作簿集合中的“三月份销售部工资管理.xlsm”（Workbook 对象）赋值给 book 变量；然后使用 If 语句判断该工作簿是否打开，如果没有打开，则执行其打开操作；最后使用 Activate 方法激活该工作簿中的“三月份考勤记录”工作表。

【提示】Workbooks.Open Filename:=ThisWorkbook.Path + “\三月份销售部工资管理.xlsm”表示打开“三月份销售部工资管理”工作簿，即“三月份考勤记录表”工作表所在的工作簿。事实上，ThisWorkbook.Path 表示当前打开的“三月份考勤记录表”工作表的路径，所以这里也可以不用这个系统变量，而是写上其绝对路径。

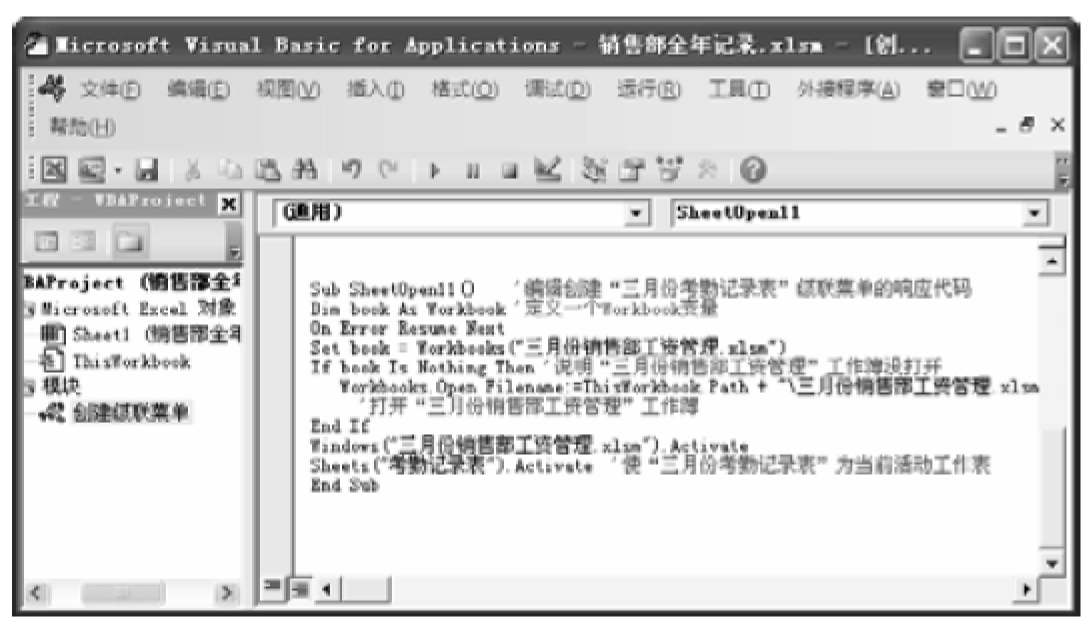


图 15-54　响应函数 SheetOpen11 的完整代码



图 15-55　响应函数 SheetOpen12 的完整代码

04　所有级联菜单的响应函数代码都很相似，这里不再详细介绍。编写的各个响应函数的代码分别如图 15-55 至图 15-60 所示。

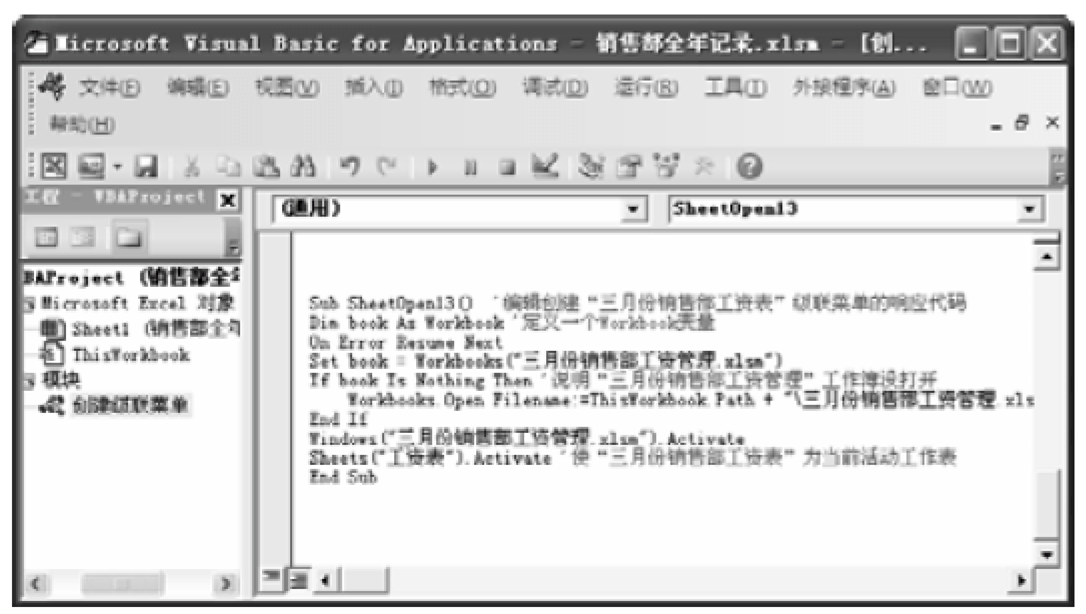


图 15-56　响应函数 SheetOpen13 的完整代码

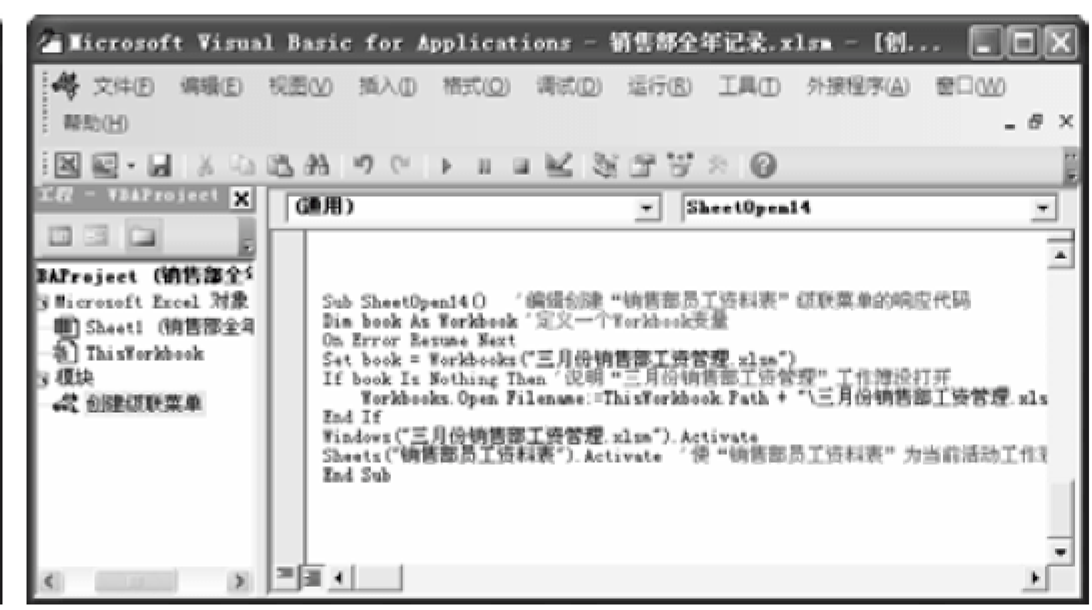


图 15-57　响应函数 SheetOpen14 的完整代码



图 15-58　响应函数 SheetOpen21 的完整代码

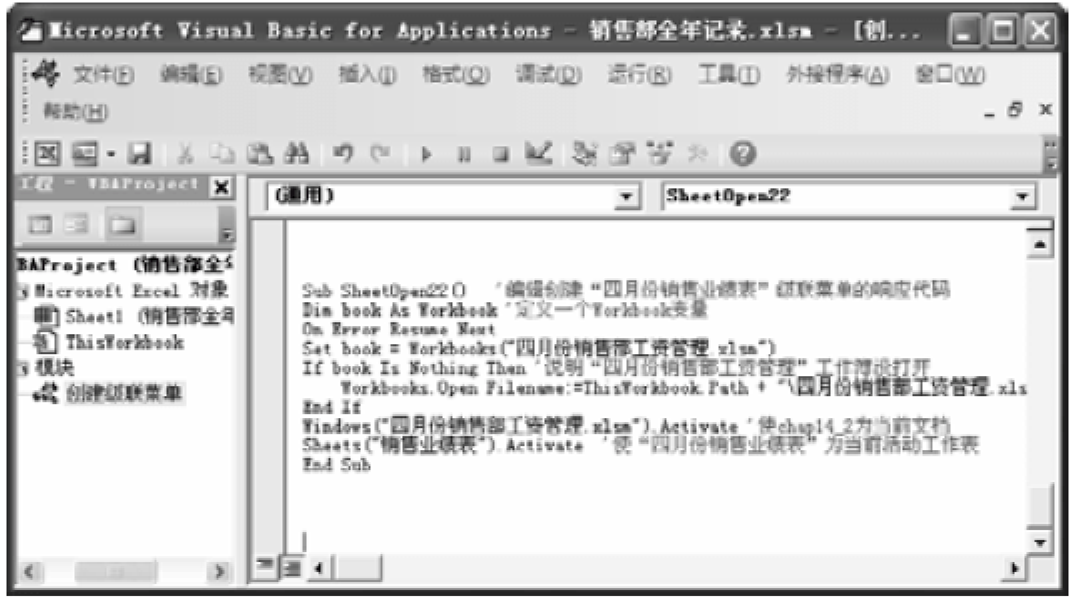


图 15-59　响应函数 SheetOpen22 的完整代码



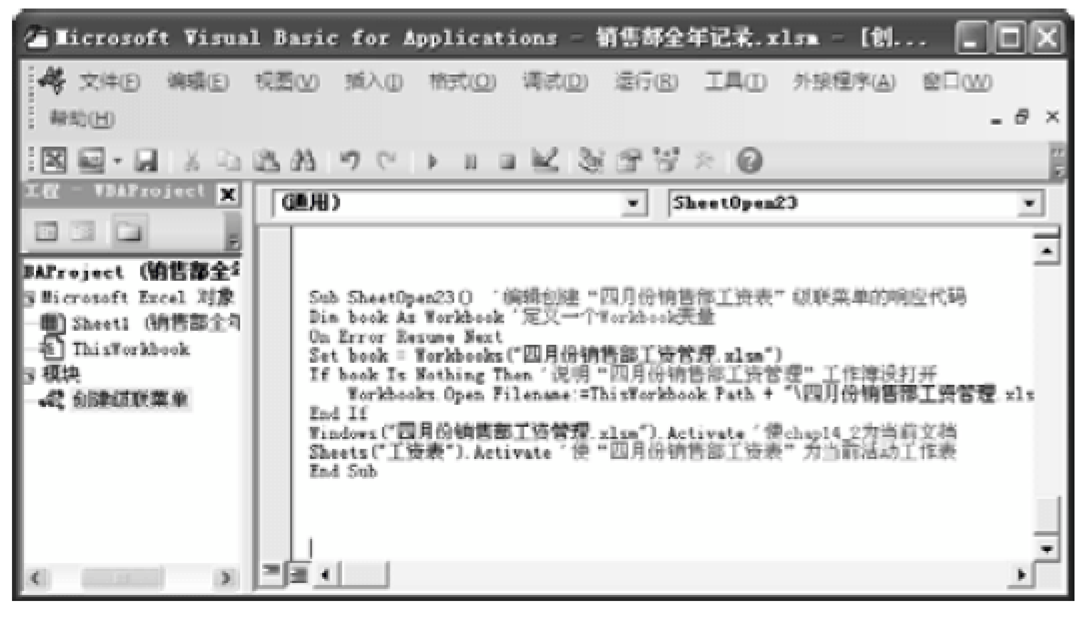


图 15-60 响应函数 SheetOpen23 的完整代码

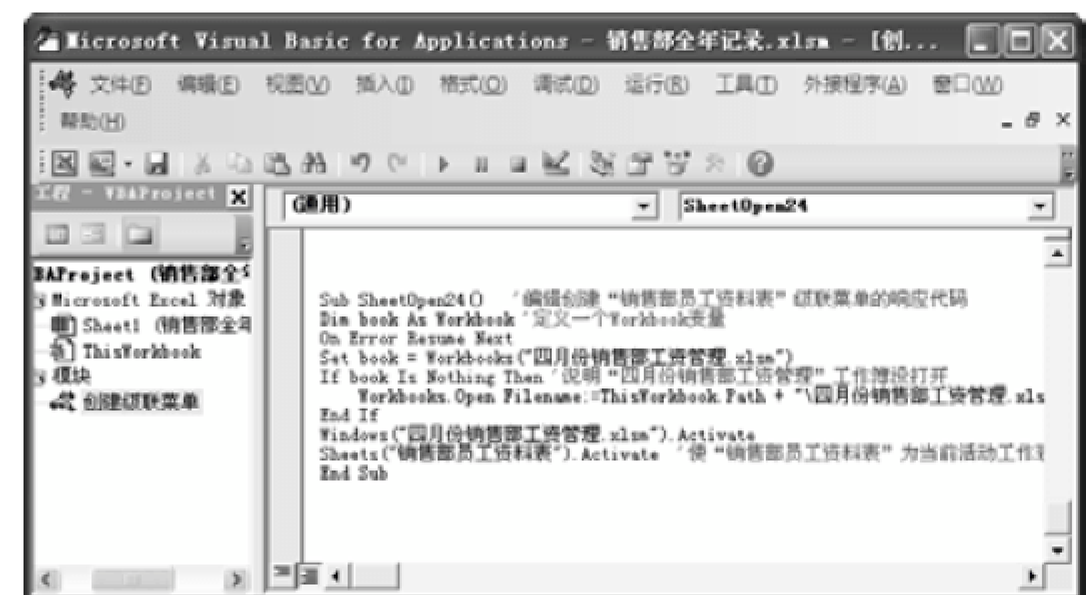


图 15-61 响应函数 SheetOpen24 的完整代码

为实现在打开“销售部全年记录”工作簿时即可使用前面创建的菜单，需要设置在该工作簿打开时运行 OpenMenu（）函数。

05 在上述代码中添加下面的打开工作簿时即可运行 OpenMenu（）函数的代码程序，如图 15-62 所示。

```
Sub auto_open() '系统自动打开运行宏
    OpenMenu '调用用户自定义菜单函数
End Sub
Sub auto_close() '系统自动关闭运行宏
    On Error Resume Next '忽略错误
    MenuBars("MyMenu").Delete '删除自定义的菜单
End Sub
```



图 15-62 打开工作簿时即可运行 OpenMenu()函数的代码

这段代码的作用是设置在“销售部全年记录”工作簿打开时即可调用 OpenMenu（）函数，而在该工作簿关闭时则自动恢复菜单为默认模式。否则，在打开其他文档时，用户都将看不到系统菜单，而只是看到这里创建的菜单项，这显然是不应该的。

15.5.3 使用级联菜单管理多个工作簿

在完成了创建多个工作簿的级联菜单后，就可以检验其运行过程了。

使用级联菜单管理多个工作簿，具体操作步骤如下。

01 重新打开“销售部全年记录”工作簿，即可看到前面创建的“销售部全年记录”工作表

成为当前活动工作表，并且所添加的 ˝三月份˝ 和 ˝四月份˝ 两个菜单项出现在【加载项】选项卡中，如图 15-63 所示。

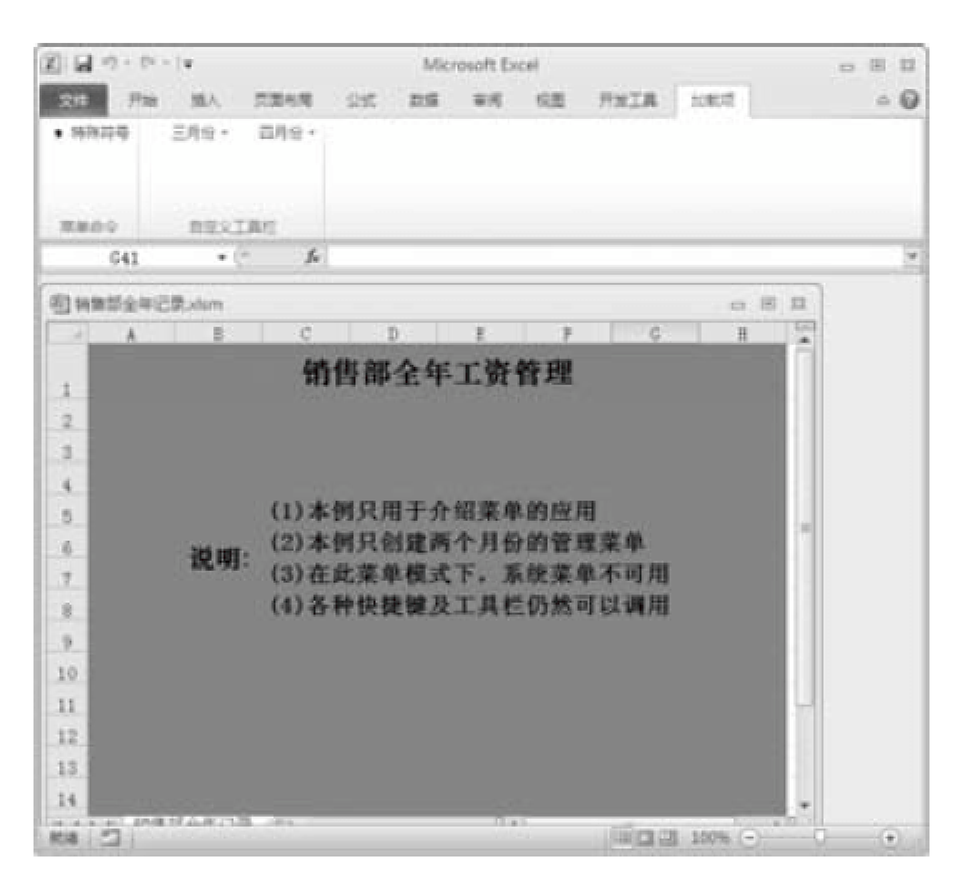


图 15-63　创建的管理多个工作簿的工作簿

02 单击【三月份】下三角按钮，即可弹出相应的级联菜单；同理，单击【四月份】下三角按钮，将弹出四月份相应的级联菜单。

03 选择【三月份】➤【三月份销售部工资表】菜单项，即可得到 ˝三月份销售部工资表˝ 成为当前活动工作表，如图 15-64 所示。

04 此时如果需要查看三月份销售部考勤记录情况，可以选择【三月份】➤【三月份考勤记录表】菜单项，或直接单击工作表标签 ˝考勤记录表˝，如图 15-65 所示。



图 15-64　“三月份销售部工资表”成为当前活动工作表　　　图 15-65　查看三月份销售部考勤记录情况

从上面的例子可以看出，使用 Excel VBA 创建管理多个工作簿的级联菜单的高效性，这里只是列举了两个月份的例子，如果需要管理全年甚至更长的时间，或是不同部门的数据，只需要增加菜单级数及增加响应函数即可。



15.6 工作表安全性保护

Excel 的应用范围非常广泛，为了防止与编辑工作表无关的人员更改其中的数据内容，需要对工作表的数据进行安全性保护。保护后的工作表数据如果未经授权，其他人都将无法对其进行修改。而当工作表中的内容不再需要保护或当编辑者需要修改其中的数据内容时，则可以取消对工作表的保护。

15.6.1 在 Excel 中实现对工作表的保护

下面以“三月份销售部工资管理”工作簿中的“工资表”为例，介绍在 Excel 中实现对工作表的保护。

1. 保护工作表

在 Excel 中实现保护“工资表”功能，具体操作步骤如下。

01 打开〝三月份销售部工资管理〞工作簿，切换到〝工资表〞。然后切换到【审阅】选项卡，在【更改】选项组中单击【保护工作表】按钮，即可打开【保护工作表】对话框，从中选中【保护工作表及锁定的单元格内容】复选框，并在〝取消工作表保护时使用的密码〞文本框中输入设置的密码，这里输入〝123456〞，如图 15-66 所示。



图 15-66 【保护工作表】对话框

【提示】在【保护工作表】对话框中，用户可以设置更为详细的保护功能，只需要在【允许此工作表的所有用户进行】列表框中选中相应的复选框即可。

02 单击【确定】按钮，系统将自动弹出【确认密码】对话框，在【重新输入密码】文本框中重新输入密码 ″123456″，如图 15-67 所示。



图 15-67　【确认密码】对话框

03 单击【确定】按钮，即可完成该工作表的保护操作。此时用户如果在该工作表的任一单元格上双击或试图修改工作表的内容或格式，系统都会弹出警告信息对话框，如图 15-68 所示。

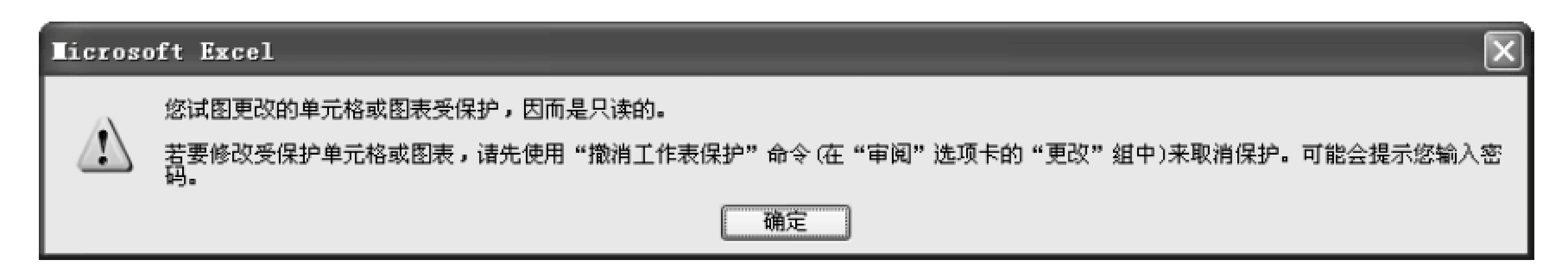


图 15-68　警告信息

04 单击【确定】按钮，即可退出该警告信息对话框。

通过以上操作可以看出，对于已经实现保护工作表功能的“工资表”，一般的用户只拥有浏览权，而无法对其进行修改；如果需要修改其中的数据内容，则必须撤销对工作表的保护。

2. 撤销工作表的保护

撤销工作表保护，具体操作步骤如下。

01 在【审阅】选项卡中，单击【更改】选项组中的【保护工作表】按钮，即可打开【撤销工作表保护】对话框，从中输入前面设置的密码 ″123456″，如图 15-69 所示。

02 单击【确定】按钮，即可取消对工作表的保护。

另外，如果同时需要对多个工作表进行保护操作，在 Excel 中只能逐个进行上述操作。这是因为如果按住 Ctrl 键的同时选定多个工作表，然后切换到【审阅】选项卡中，会发现在【更改】选项组中【保护工作表】按钮处于不可用状态，而只能选择【保护工作簿】菜单项，如图 15-70 所示。



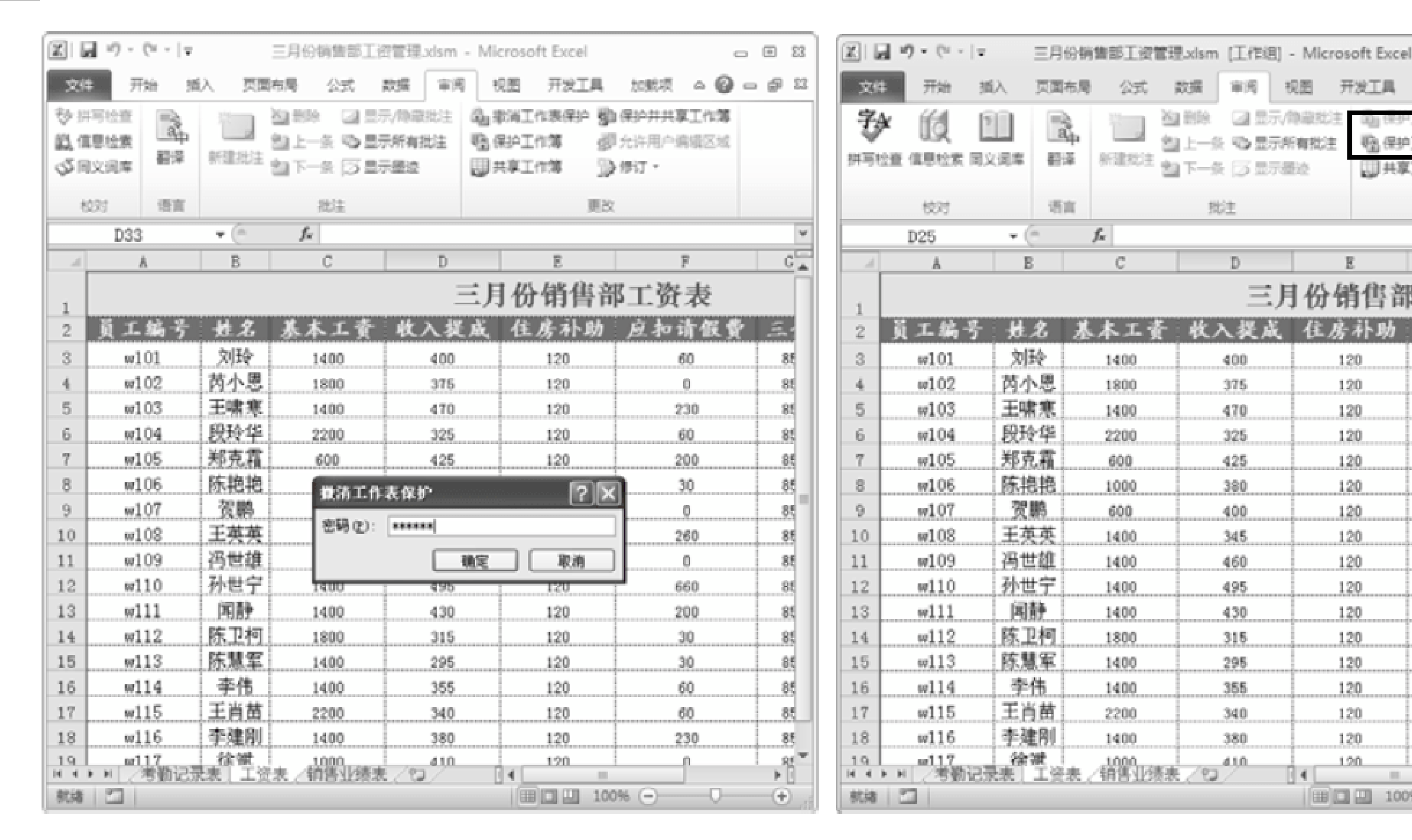


图 15-69 【撤销工作表保护】对话框　　　　图 15-70 【更改】选项组

15.6.2 使用 Excel VBA 实现对多个工作表的保护

使用 Excel VBA 可以实现对多个工作表的保护功能。下面将设计一个保护多个工作表的窗体，该窗体由“保护”和“打开”两部分组成。在“保护”部分列举出当前工作簿中所有未被保护的工作表，当用户选择其中任意一个工作表时，对该工作表执行加密操作；在“打开”部分列举出当前工作簿中已经被保护的工作表，当用户选择其中任意一个工作表时，对其执行解密操作，从而实现多个工作表的安全性保护。

1. 创建保护多个工作表的窗体

在“三月份销售部工资管理”工作簿中使用 Excel VBA 创建一个保护多个工作表的窗体，具体操作步骤如下。

01 在“三月份销售部工资管理”工作簿中，切换到【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮，即可进入 VBA 代码窗口，从中选择【插入】➤【用户窗体】菜单项，在该项目工程中插入一个用户窗体。接着将其【属性】窗口中的 Caption 属性修改为“打开并保护多个工作表”，如图 15-71 所示。

02 单击【工具箱】中的【标签】按钮 A，在用户窗体中绘制一个标签控件。接着在其【属性】窗口中的 Caption 文本框中输入“请选择要保护的工作表”，按下 Enter 键即可修改完成其显示文本，如图 15-72 所示。

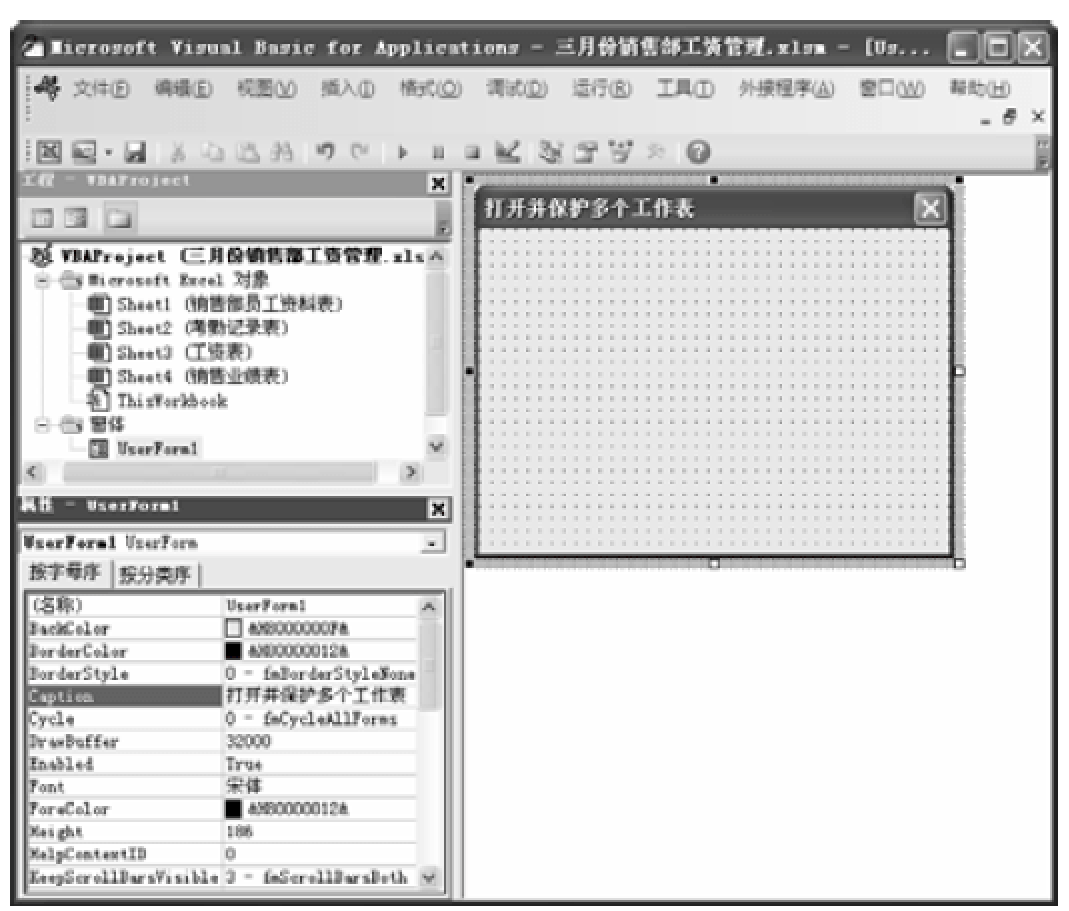


图 15-71　设置用户窗体的 Caption 属性值

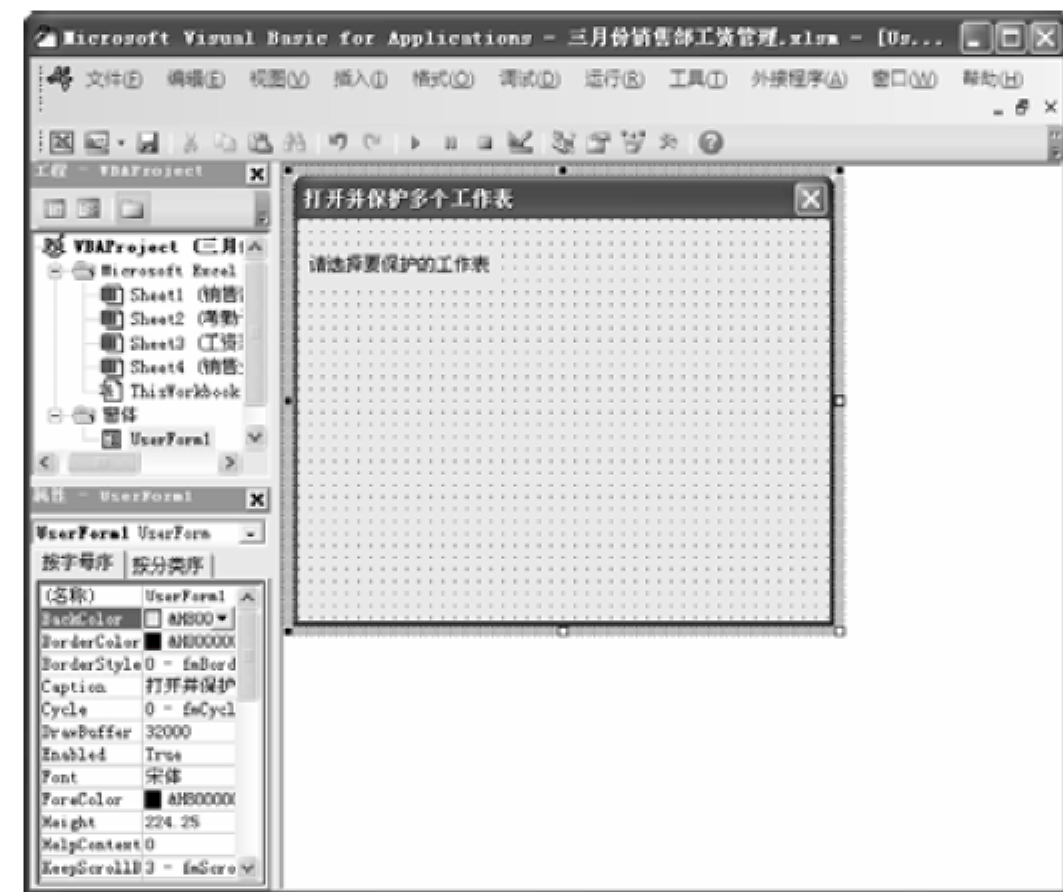


图 15-72　添加一个标签控件

03 单击【工具箱】中的【组合框】按钮，在用户窗体中绘制一个组合框控件，如图 15-73 所示。

04 单击【工具箱】中的【列表框】按钮，在用户窗体中绘制一个列表框控件，如图 15-74 所示。

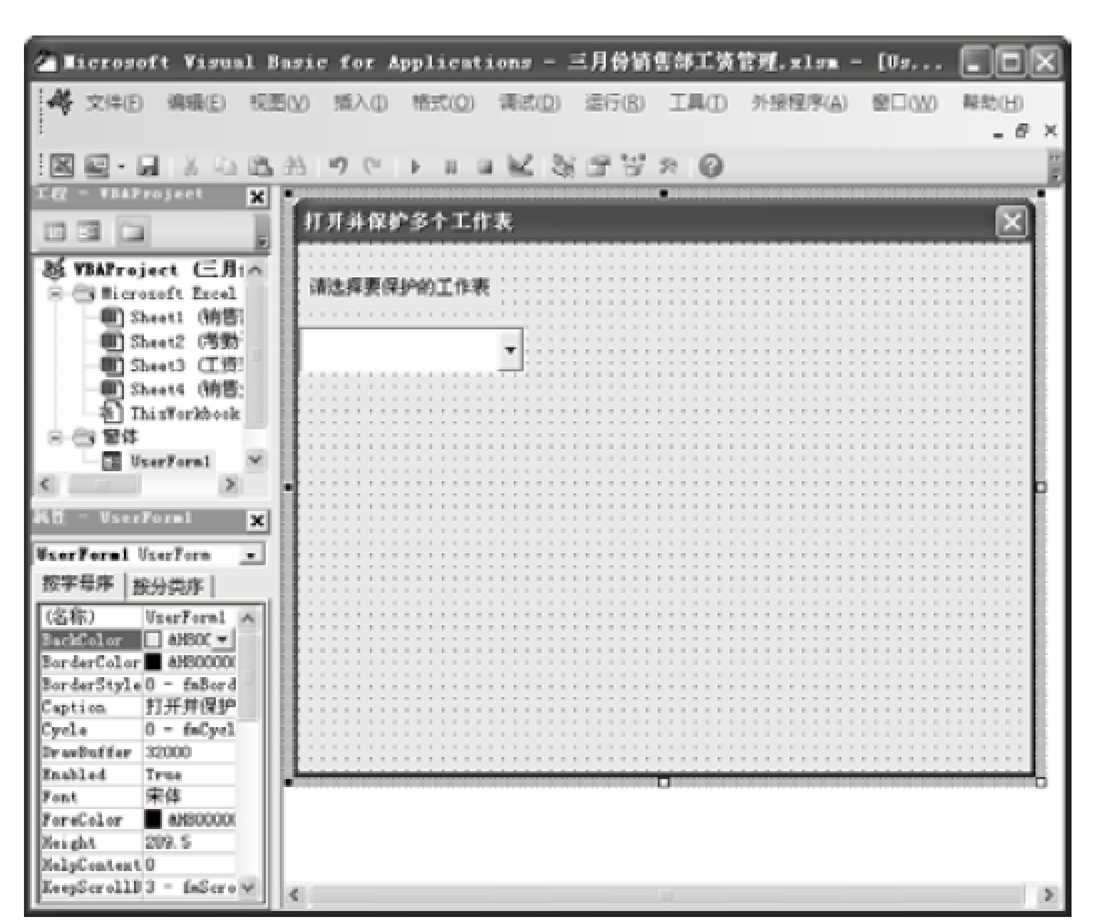


图 15-73　添加一个组合框控件

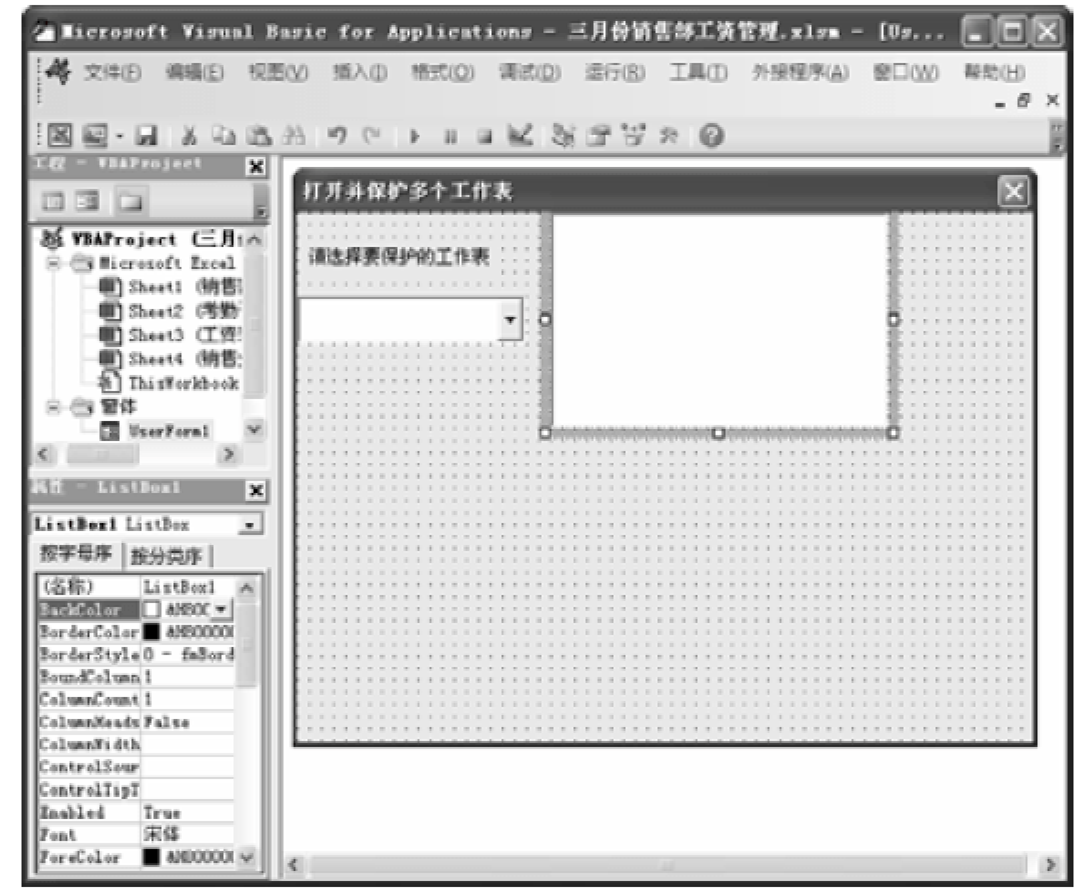


图 15-74　添加一个列表框控件

05 单击【工具箱】中的【命令按钮】，在用户窗体中绘制一个命令按钮控件，并在其【属性】窗口中的 Caption 文本框中输入“保护”，即修改添加按钮的显示文本，如图 15-75 所示。

06 按照上面的操作步骤，在【打开并保护多个工作表】用户窗体中添加如图 15-76 所示的内容。



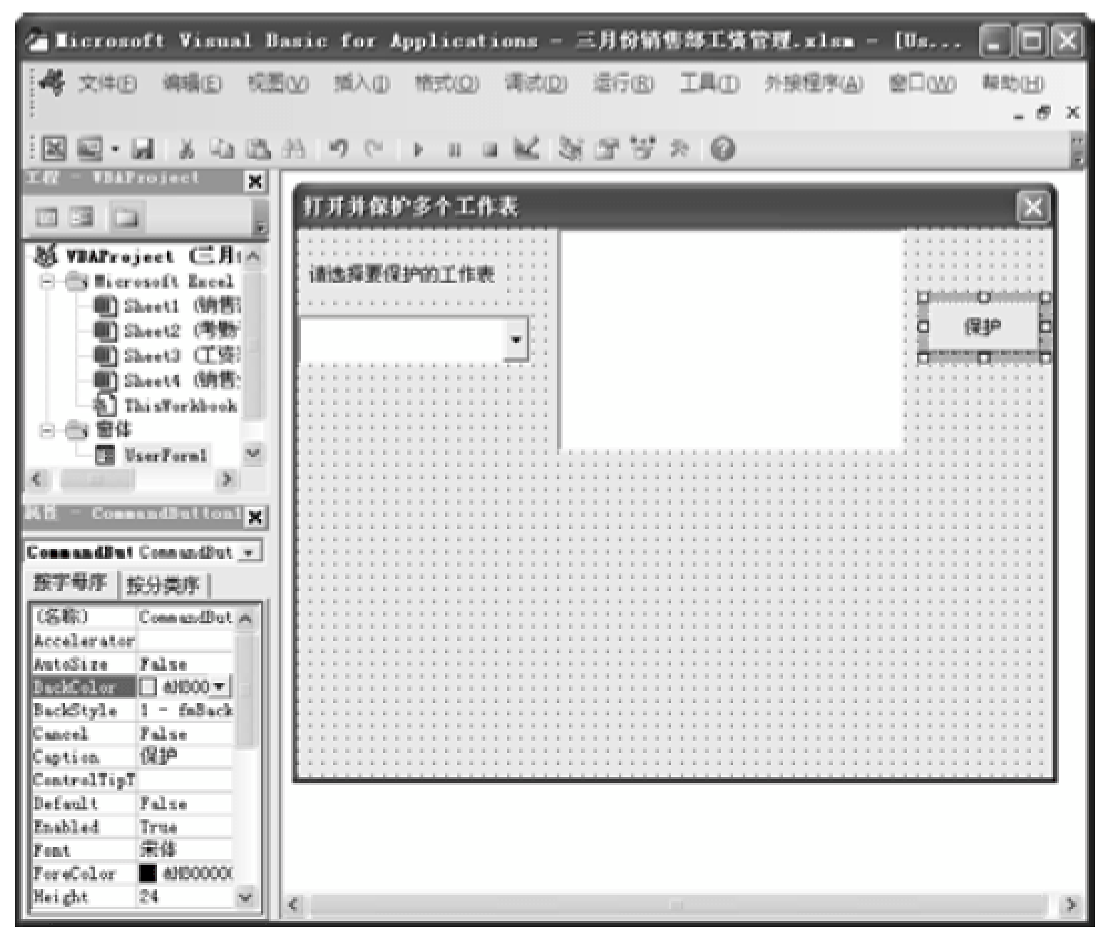


图 15-75　添加一个命令按钮控件

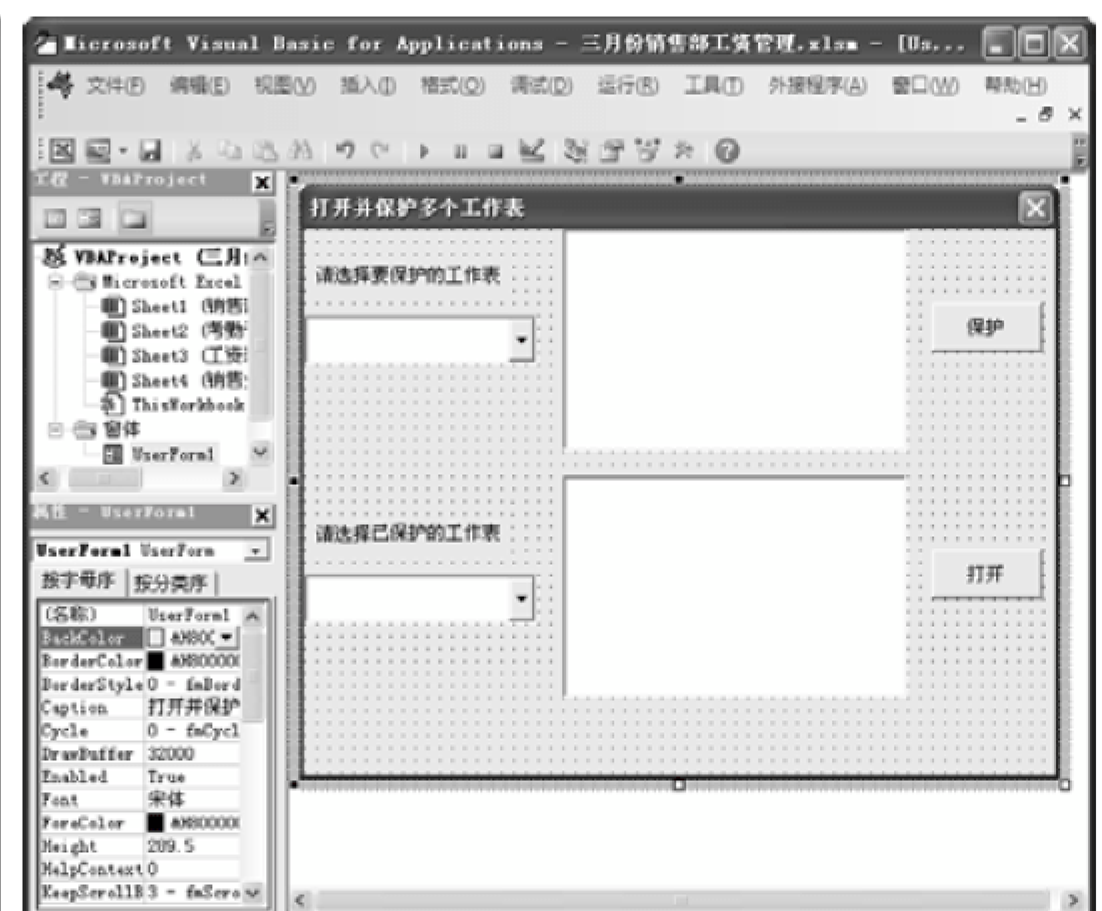


图 15-76　设计完成的用户窗体

2. 编辑保护多个工作表的 VBA 代码

创建好一个保护多个工作表的窗体后，就可以编辑实现保护多个工作表功能的 VBA 代码了。在 VBA 中实现保护多个工作表，具体的操作步骤如下。

01 双击【打开并保护多个工作表】窗体，进入其代码编辑窗口。然后在其对象下拉列表中选择 UserForm 选项，在其事件下拉列表中选择 Activate 选项，即可生成 UserForm_Click 事件处理程序代码框架，如图 15-77 所示。

02 在该代码框架中输入下面的代码程序，如图 15-78 所示。

```
Private Sub UserForm_Activate()
Dim book As Worksheet    '定义一个工作表变量
For Each book In ActiveWorkbook.Sheets
    '获得当前工作簿中工作表的名称
    If book.ProtectContents = False Then
        ComboBox1.AddItem book.Name
         '定义加载第一个组合框时显示的内容
    Else
        ComboBox2.AddItem book.Name
         '定义加载第二个组合框时显示的内容
    End If
Next book
End Sub
```



图 15-77　系统自动生成的代码框架

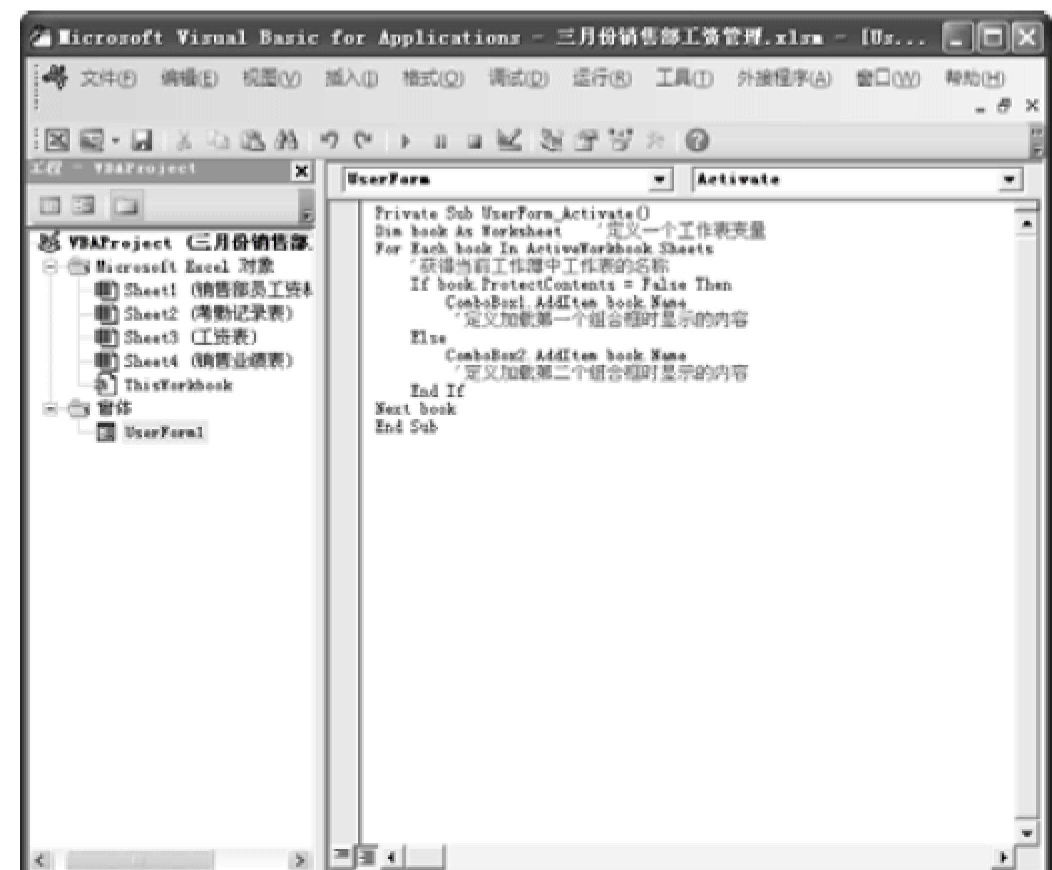


图 15-78　编辑 UserForm_Click 的事件处理程序

这段代码中的“ActiveWorkbook.Sheets”表示当前工作簿中所有工作表的集合，“book.ProtectContents”表示选择的工作表是否处于保护状态。其具体实现过程是，首先定义一个工作表数据类型的变量 book，然后在循环语句中使工作表变量 book 依次取得该工作表集合中每个工作表的名字，接着使用 If 语句判断该工作表是否处于保护状态，如果未被保护，则将其添加到组合框 ComboBox1 中；否则将添加到组合框 ComboBox2 中。

03 代码输入完毕后，选择【运行】➤【运行子过程/用户窗体】菜单项，然后在弹出的【打开并保护多个工作表】窗体中单击【请选择要保护的工作表】下三角按钮，可弹出如图 15-79 所示的下拉列表。

04 单击【请选择已保护的工作表】下三角按钮，则可以看到下拉列表中没有任何内容，如图 15-80 所示。这是由于此时该工作簿中所有的工作表都未被保护。

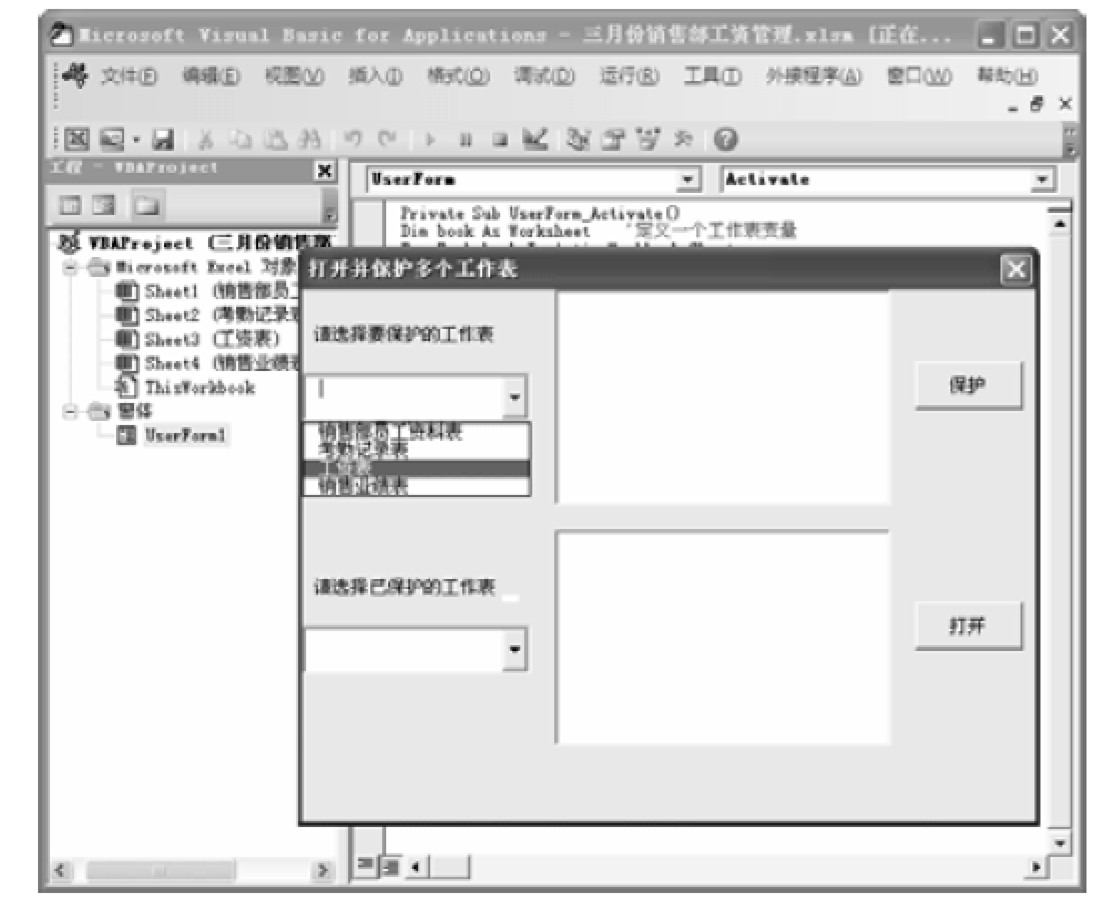


图 15-79　【请选择要保护的工作表】下拉列表

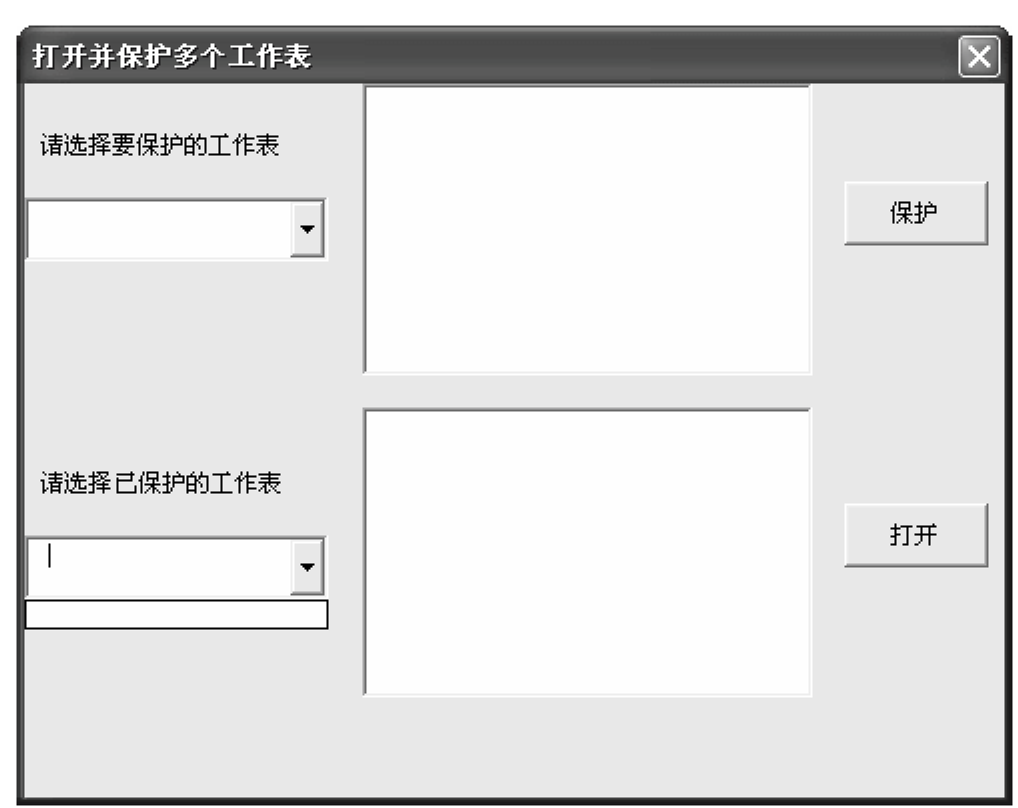


图 15-80　【请选择已保护的工作表】下拉列表

05 如果用户在下拉列表中选择任意一个工作表，在右侧的列表框中都将显示出该工作表的名称，并且列表框中的工作表也就是需要加密或者解密的工作表。所以，还需要进一步设置组合

框的 Change 事件处理程序，即双击添加的第一个组合框 ComboBox1，进入其代码编辑窗口，输入如下代码：

```
Private Sub ComboBox1_Change()
ListBox1.AddItem ComboBox1.Text '设置列表框 1 的显示内容
End Sub
```

这段代码的作用是当用户在组合框 ComboBox1 中选择了一个工作表时，系统将自动加载列表框 ListBox1 的显示内容。按照同样的方法，编辑组合框 ComboBox2 的 Change 事件处理程序，即双击第二个组合框，进入其代码编辑窗口，输入下面的代码程序：

```
Private Sub ComboBox2_Change()
ListBox2.AddItem ComboBox2.Text '设置列表框 2 的显示内容
End Sub
```

06 此时，在 VBA 代码编辑窗口中，选择【运行】→【选择子过程/用户窗体】菜单项，在弹出的【打开并保护多个工作表】用户窗体中，在【请选择要保护的工作表】下拉列表中选择"销售部员工资料表"选项，即可在其下面的列表框中自动显示出选择的工作表名称"销售部员工资料表"，如图 15-81 所示。

07 有时，用户可能误单击了组合框中的选项，将不需要保护的工作表添加到列表框中，可以通过设置列表框的双击事件来删除这些不需要的项目，即在代码窗口中输入下面的 ListBox1_DblClick 事件处理程序的代码。

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
ListBox1.RemoveItem ListBox1.ListIndex
  '设置当双击列表框的项目时系统自动删除当前项目
End Sub
```

这段代码的作用是每次双击列表框中显示的内容时，系统都会自动将其删除。其中 ListBox1.RemoveItem 是删除列表框中当前内容的方法；ListBox1.ListIndex 是列表框中当前选中项目的序列号 Index。

08 按照同样的方法设置列表框 ListBox2 的双击事件，即在其双击事件处理程序代码框架中输入如下代码：

```
Private Sub ListBox2_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
ListBox2.RemoveItem ListBox2.ListIndex
  '设置当双击列表框的项目时系统自动删除当前项目
End Sub
```

09 此时，运行【打开并保护多个工作表】用户窗体，在打开的【打开并保护多个工作表】用户窗体中，双击列表框中自动出现的"销售部员工资料表"选项，系统即可自动删除该列表框中的显示内容，如图 15-82 所示。



图 15-81　在列表框中显示用户选择的下拉列表选项



图 15-82　删除列表框中显示的内容

【提示】双击事件是将列表框中显示的内容删除掉，但是在其下拉列表中仍然包含该项目内容，如图 15-83 所示。



图 15-83　下拉列表保留项目

10 在 VBA 代码窗口中，双击【打开并保护多个工作表】窗体中的【保护】按钮，进入其代码编辑窗口，在该代码窗口中输入如下代码，如图 15-84 所示。

```
Private Sub CommandButton1_Click()
Dim pwd As String, i As Integer '定义一个用来获得密码的字符串变量pwd
If ComboBox1.Text <> "" Then
    pwd = InputBox("请输入保护密码", "加密") '输入保护密码
    If pwd = "" Then
        Exit Sub '密码为空则直接退出
    End If
    For i = 0 To ListBox1.ListCount - 1
        Sheets(ListBox1.List(i)).Protect pwd '对列表框中显示的所有表格加密
    Next i
    MsgBox "加密成功！"
Else
    MsgBox "请选择需要保护的工作表！"
    Exit Sub
```

```
End If
End Sub
```

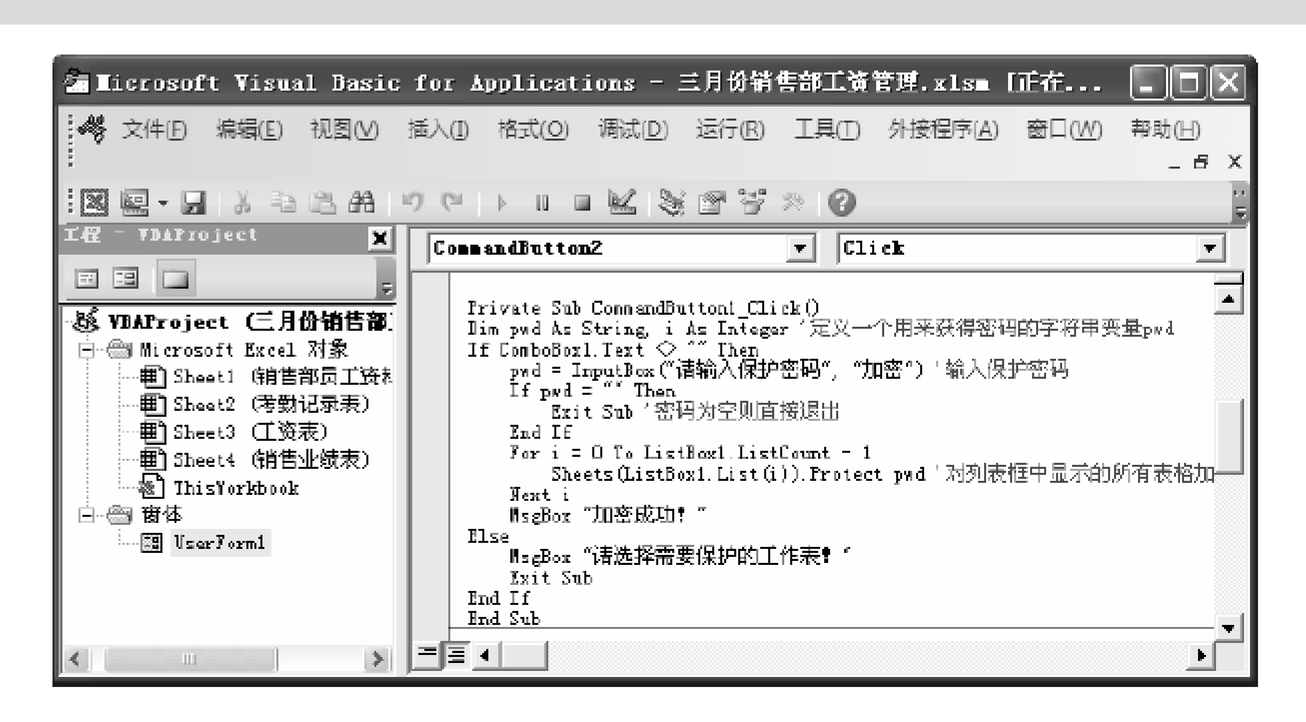


图 15-84 【保护】按钮的响应程序

这段代码是使用 If 语句判断当前组合框中是否有工作表信息，如果有，则首先使用 InputBox 方法获取保护密码，然后再使用 Sheets（ListBox1.List（i））.Protect pwd 语句对列表框中显示的所有工作表进行保护操作。同理，也可以使用 Sheets（ListBox1.List（i））.Unprotect pwd 语句对已经被保护的工作表撤销保护。

11 按照同样的方法，设置【打开】按钮的事件处理程序，即在其代码窗口中输入下面的程序代码，如图 15-85 所示。

```
Private Sub CommandButton2 Click()
Dim pwd As String, i As Integer '定义一个用来获得密码的字符串变量 pwd
If ComboBox2.Text = "" Then
    MsgBox "请选择已经受保护的工作表！"
    Exit Sub
Else
    pwd = InputBox("请输入解密密码", "解密") '输入解密密码
    If pwd = "" Then Exit Sub '密码为空则直接退出
    For i = 0 To ListBox2.ListCount - 1
      Sheets(ListBox2.List(i)).Unprotect pwd '对列表框中显示的所有表格解密
    Next i
    MsgBox "解密成功！"
End If
End Sub
```

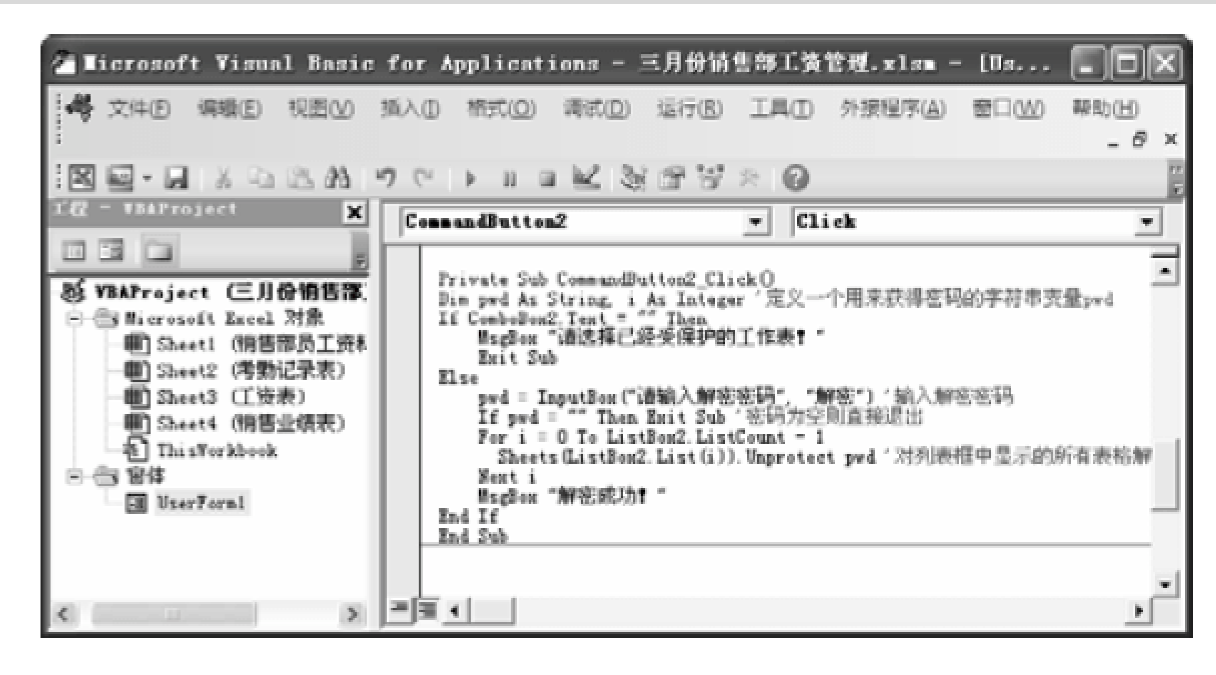


图 15-85 【打开】按钮的响应程序

到此，就完成了保护多个工作表的 VBA 代码的编辑过程。为了快捷方便地运行该 VBA 代码，可以为其设置运行方式。

3. 运行 VBA 代码打开并保护多个工作表

完成了使用 Excel VBA 设计【打开并保护多个工作表】窗体的过程后，就可以运行该窗体，检验其功能。

使用 VBA 代码保护工作表，具体的操作步骤如下。

01 在 VBA 代码窗口中，选择【运行】➤【运行子过程/用户窗体】菜单项，即可弹出【打开并保护多个工作表】窗体，在【请选择要保护的工作表】下拉列表中显示了所有未被保护的工作表，在其中选择 "考勤记录表" 选项，即可在其右侧上方列表框中显示所选内容，如图 15-86 所示。



图 15-86　选择“考勤记录表”选项

【提示】此时若不想保护该“考勤记录表”工作表，只需双击该选项即可将其从列表框中删除。

02 选中列表框中的 "考勤记录表" 工作表，然后单击【保护】按钮，系统将自动弹出【加密】对话框，输入保护密码 "123456"，如图 15-88 所示。

03 单击【确定】按钮，系统将自动弹出加密成功的提示信息，如图 15-89 所示。







图 15-88　【加密】对话框　　图 15-89　提示加密成功

04 单击【确定】按钮，退出该对话框，然后关闭【打开并保护多个工作表】对话框，返回 "考勤记录表" 工作表中，此时如果双击该工作表上的任意单元格或试图修改该工作表，系统则会弹出如图 15-90 所示的警告信息。

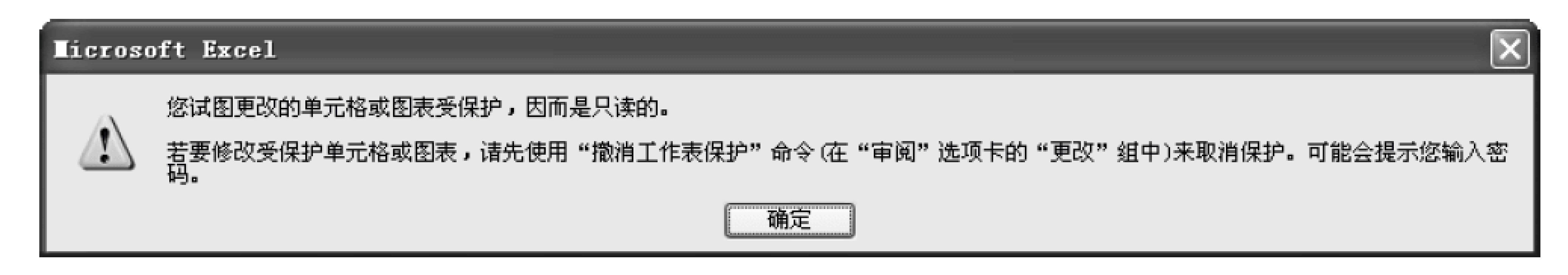


图 15-90　警告信息

05 在 VBA 代码编辑窗口中再次运行【打开并保护多个工作表】用户窗体，此时在【请选择已保护的工作表】下拉列表中即可显示已被保护的“考勤记录表”工作表，如图 15-91 所示。

06 接着选择【考勤记录表】选项，即可在其右侧下方列表框中显示出所选结果，如图 15-92 所示。



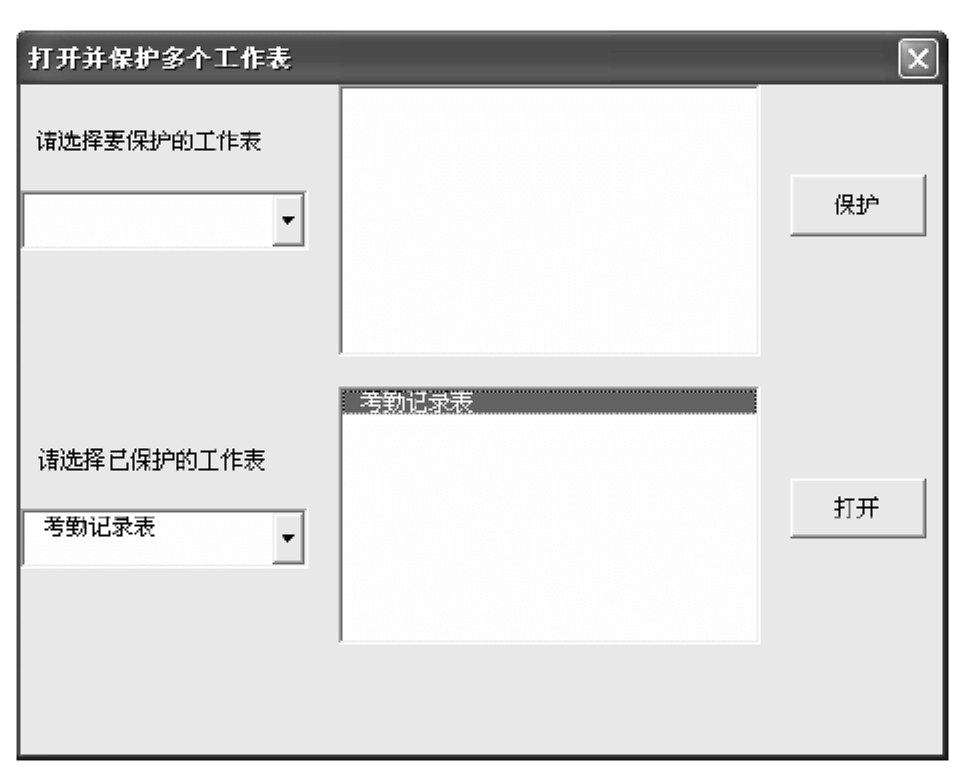


图 15-91　显示已被保护的工作表　　图 15-92　选择“考勤记录表”选项

07 单击【打开】按钮，系统将自动弹出【解密】对话框，在【请输入解密密码】文本框中输入密码“123456”，如图 15-93 所示。

08 单击【确定】按钮，系统将自动弹出解密成功的提示信息，如图 15-94 所示。此时返回到“考勤记录表”工作表中，将会看到该工作表已被撤销保护。

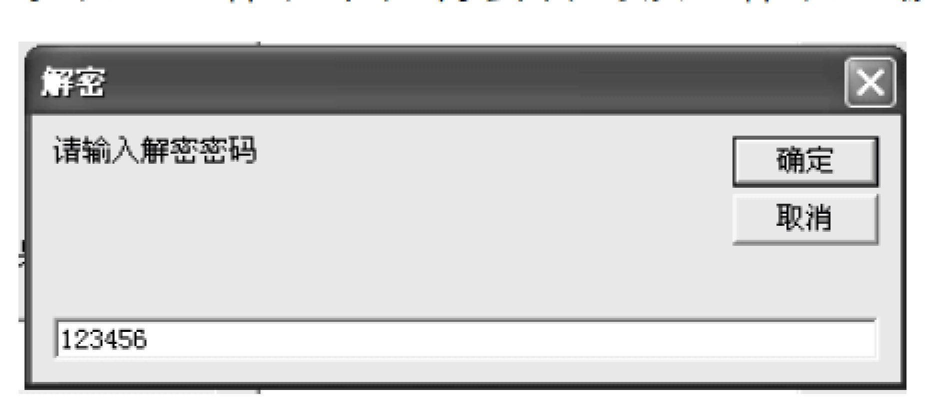






图 15-93　【解密】对话框　　图 15-94　提示解密成功

09 如果在【解密】对话框中输入了错误的解密密码，如图 15-95 所示，单击【确定】按钮后，系统将会弹出一个错误提示信息对话框，如图 15-96 所示。

另外，如果用户没有选择需要保护的工作表，而直接单击【保护】按钮，系统将弹出“请选择需要保护的工作表”提示信息，如图 15-97 所示；如果用户没有选择需要打开的工作表，而直接单击【打开】按钮，系统将自动弹出“请选择已经受保护的工作表”提示信息，如图 15-98 所示。



图 15-95　输入错误的解密密码



图 15-96　提示密码错误



图 15-97　提示选择需要保护的工作表



图 15-98　提示选择受保护的工作表



15.7 综合实战——使用 VBA 综合操作文件

文件的操作包括创建文件、读取文件中的数据和获取文件属性信息。本章的综合实战将包含实现这些操作的 VBA 代码。具体操作步骤如下。

01 打开一个 Excel 表单，在 Sheet1 表单中插入一个文本框和按钮控件，如图 15-99 所示。

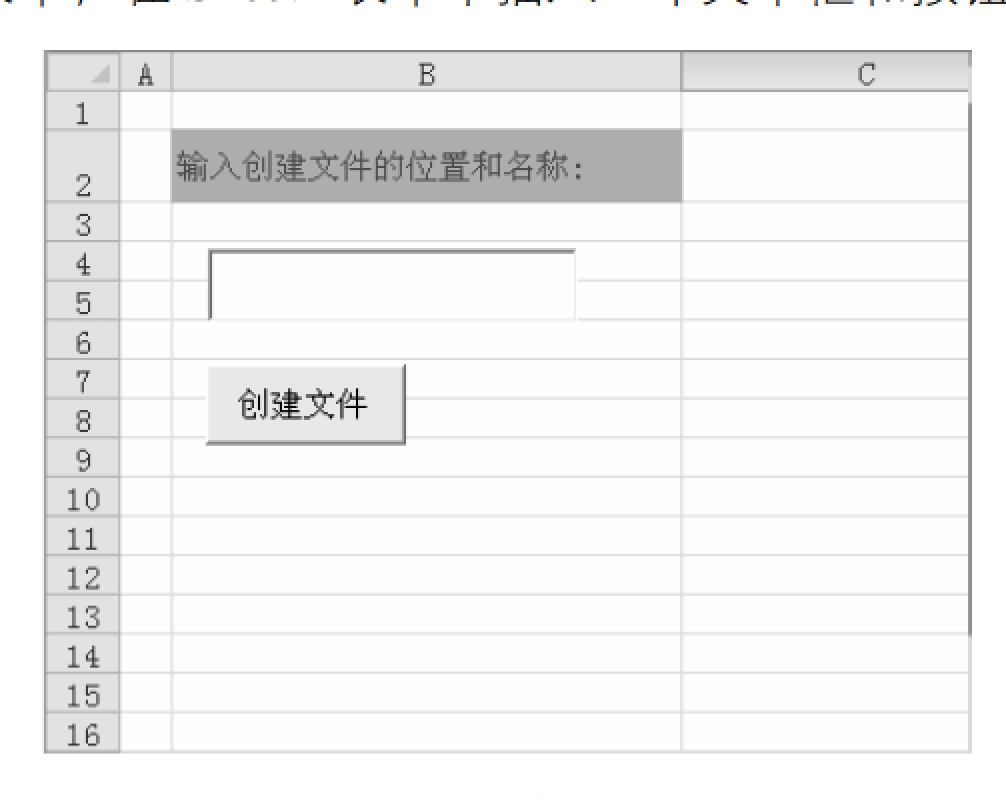


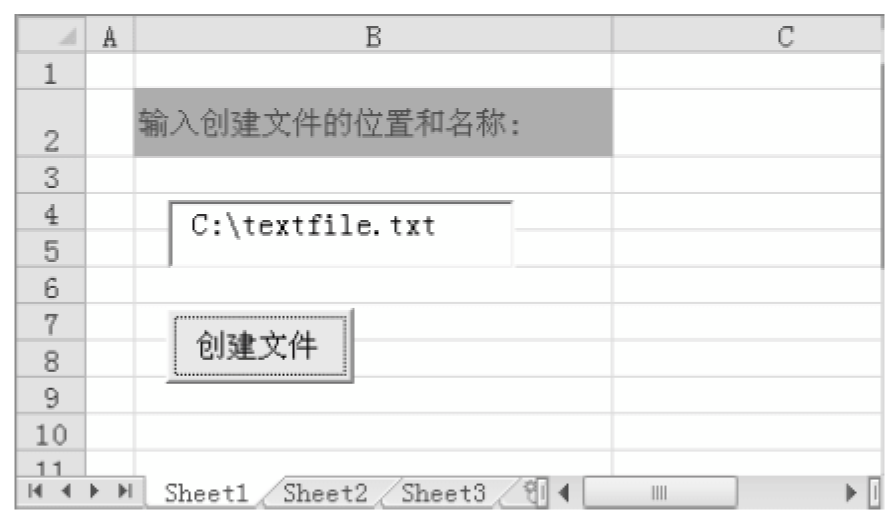
图 15-99　插入文本框和按钮控件

02 双击【创建文件】按钮控件，进入 Sheet1 表单的代码编写窗口，输入代码如下：

```
Private Sub CommandButton1_Click()
  Dim fso As New FileSystemObject
  Dim mystr As String
  mystr = TextBox1.Text
  If mystr = "" Then
    MsgBox "请输入文件位置及名称!", , "提示对话框"
  Else
```

```
        Set fill = fso.CreateTextFile(mystr, True)
        MsgBox "成功创建文件,文件位置及名称为 " + mystr, , ""
        fill.Close
    End If
End Sub
```

03 代码输入完毕，返回 Excel 表单，在文本框中输入要创建的文件的位置和名称，输入完成之后，单击【确定】按钮，执行文件创建操作，如图 15-100 所示。创建成功之后，将弹出提示信息，如图 15-101 所示。



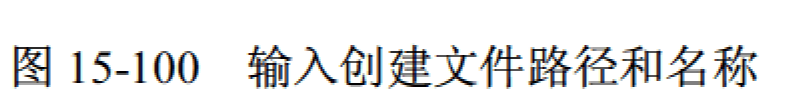


图 15-100 输入创建文件路径和名称



图 15-101 创建成功消息

04 打开 Sheet2 表单，插入文本和按钮控件，如图 15-102 所示。

05 在 textfile.txt 文件中输入一些表格数据，如图 15-103 所示。

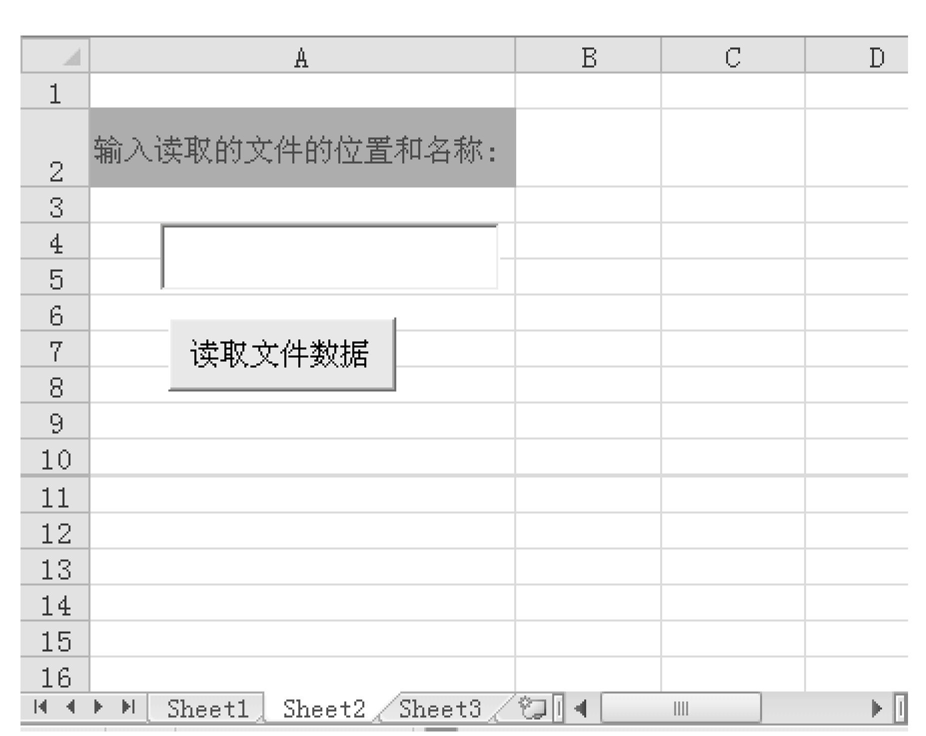


图 15-102 插入读取文件数据控件



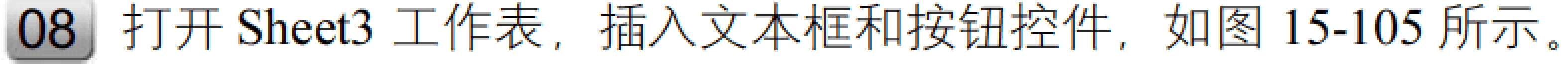
图 15-103 向文件中插入数据

06 双击【读取文件数据】按钮控件，进入 Sheet2 表单的代码编写窗口，输入代码如下：

```
Private Sub CommandButton1_Click()
Dim Filename As String
Dim x As Long, y As Integer  '相对于活动单元格的偏移量
Dim str As String, temp As String
Dim Data As Variant
Dim i As Integer
On Error Resume Next  '忽略错误，执行错误语句之后的语句
Filename = " D:\Employee\员工信息.txt "
Open Filename For Input As #1
If Err <> 0 Then
    MsgBox "无法找到 " & Filename, vbCritical, "ERROR"
    Exit Sub
```

```
    End If
    x = 0
    y = 0
    str = ""
    Application.ScreenUpdating = False    '忽略屏幕刷新
    Do Until EOF(1)
        Line Input #1, Data
        For i = 1 To Len(Data)
            temp = Mid(Data, i, 1)  '获取 Data 中的第 i 个字符
            If temp = "," Then  '以逗号为分隔符
                ActiveCell.Offset(x, y) = str
                y = y + 1
                str = ""
            Else
                str = str & temp
                If i = Len(Data) Then  '文本文件中一行的最后一个字符
                    ActiveCell.Offset(x, y) = str
                    str = ""
                    Exit For
                End If
            End If
        Next i
        y = 0
        x = x + 1
    Loop
    Close #1  '关闭文本文件
    End Sub
```

07 返回 Sheet2 工作表中，单击【控件】选项组中的【退出设计模式】按钮退出命令按钮的设计模式。首先选中一个单元格，这里选中 B6 单元格，然后单击【读取文件数据】按钮，即可将 textfile.txt 文本文件中的数据存放在以活动单元格 B6 开始的单元格中，如图 15-104 所示。



08 打开 Sheet3 工作表，插入文本框和按钮控件，如图 15-105 所示。

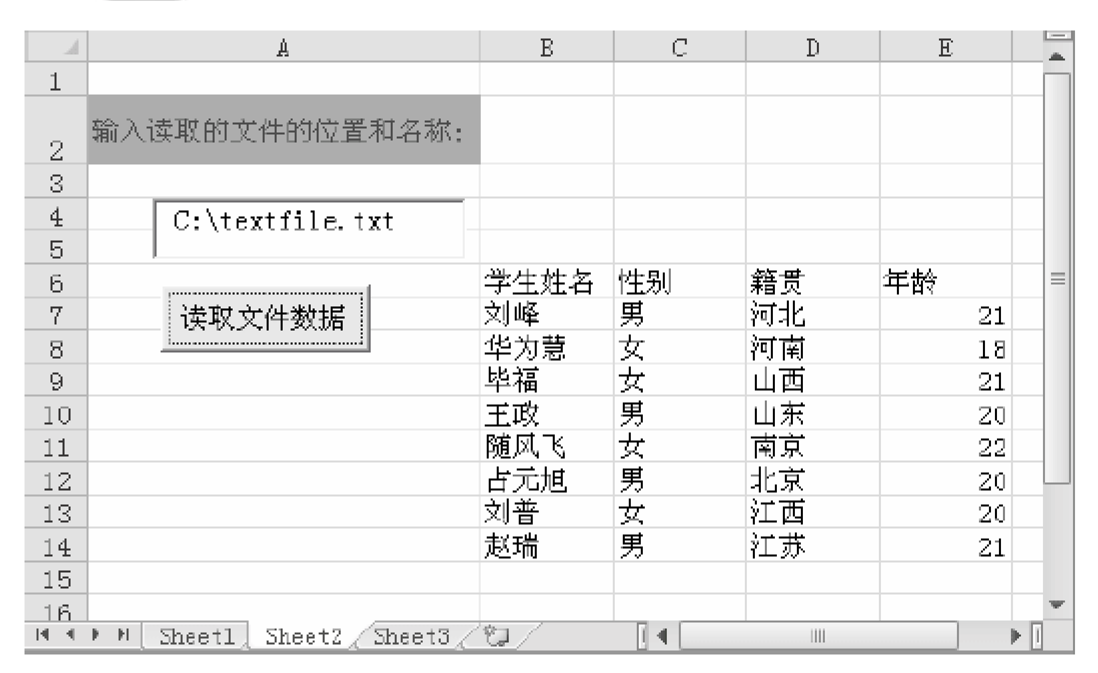


图 15-104　读取文本文件中的数据

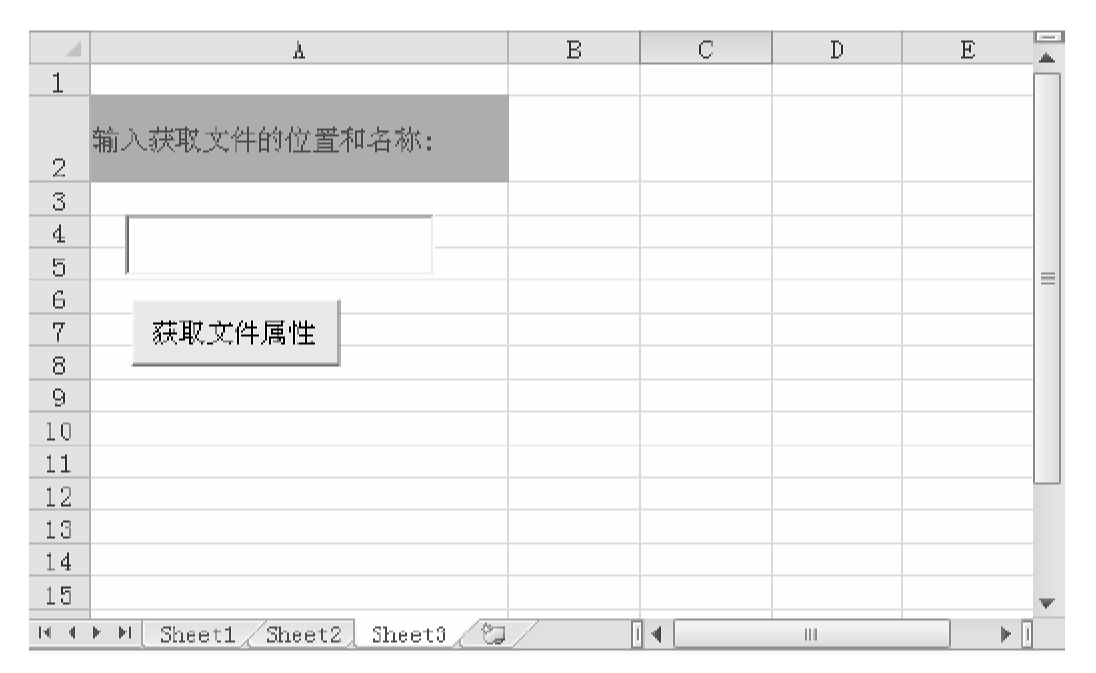


图 15-105　插入获取文件属性控件

09 双击【获取文件属性】按钮控件，进入 Sheet3 工作表的代码编写窗口，输入代码如下：

```
Private Sub CommandButton1_Click()
  Dim fso As New FileSystemObject
  Dim fill As File
  Dim mystr  As String
  Dim myfilename As String
```




```
    myfilename = TextBox1.Text
    If fso.FileExists(myfilename) Then
     Set fill = fso.GetFile(myfilename)
     mystr = ""
     mystr = mystr & "文件名： " & fill.Name & vbCrLf
     mystr = mystr & "文件大小： " & fill.Size & vbCrLf
     mystr = mystr & "文件创建时间： " & fill.DateCreated & vbCrLf
     mystr = mystr & "文件最后访问时间： " & fill.DateLastAccessed & vbCrLf
     mystr = mystr & "文件最后修改时间： " & fill.DateLastModified & vbCrLf
     mystr = mystr & "文件路径： " & fill.Path & vbCrLf
     mystr = mystr & "文件所在的目录： " & fill.ParentFolder & vbCrLf
     mystr = mystr & "文件类型： " & fill.Type & vbCrLf


      Cells(9, 1) = "文件名称"
      Cells(9, 2) = "文件类型"
      Cells(9, 3) = "大小"
      Cells(9, 4) = "修改日期"
      Cells(9, 5) = "文件路径"

      Cells(10, 1) = fill.Name    '文件的名称
      Cells(10, 2) = fill.Type 'fso.GetFile(Directory & s).Type '文件的类型
      Cells(10, 3) = fill.Size & "bytes"  'Trim(str(Int(FileLen(Directory & s)
/ 1024) + 1)) & " KB"
      Cells(10, 4) = fill.DateLastModified  '修改时间
      Cells(10, 5) = fill.Path
    Else
     MsgBox "你要查看的文件不存在,请重新输入!"
    End If
  End Sub
```

10 返回 Sheet3 工作表中，单击【控件】选项组中的【退出设计模式】按钮退出命令按钮的设计模式。首先选中一个单元格，这里选中 B6 单元格，然后单击【获取文件属性】按钮，即可将 textfile.txt 文本文件的属性信息存放在以活动单元格 A9 开始的单元格中，如图 15-106 所示。

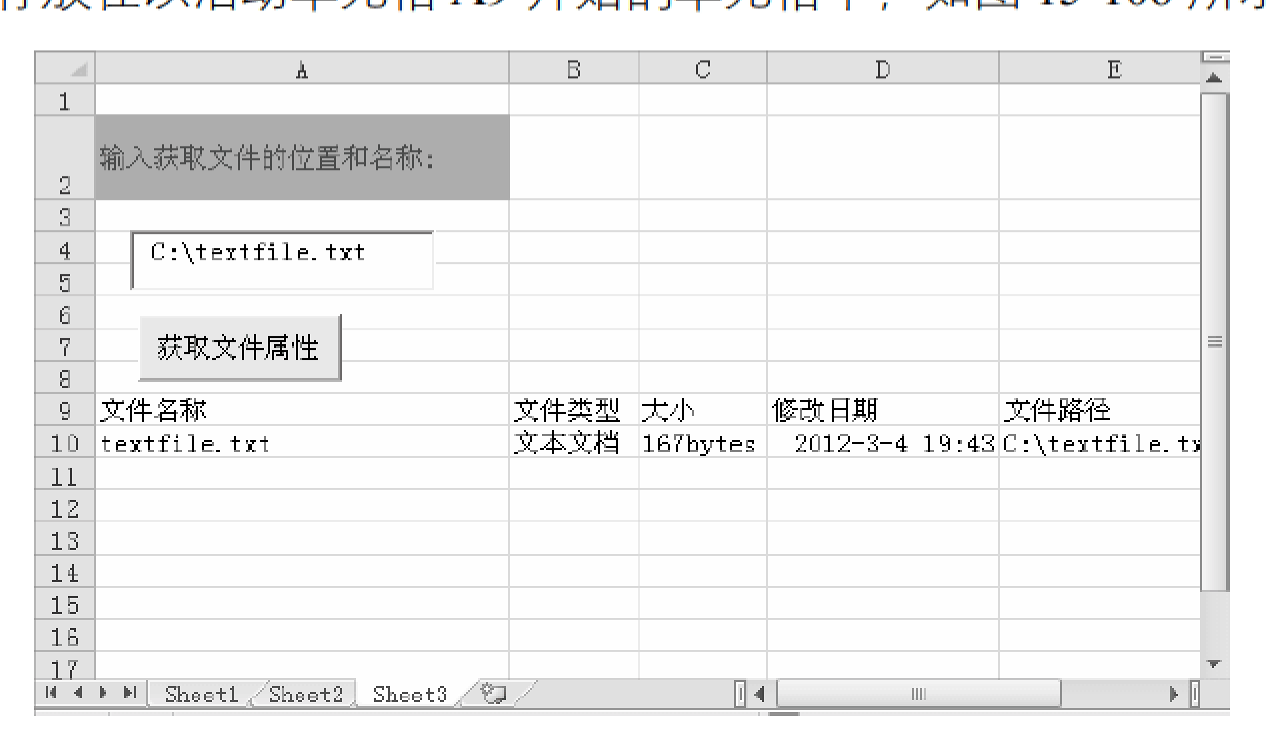


图 15-106　获取文件属性信息



15.8 高手私房菜

技巧 1：解决无法导入文本的问题

在将文本导入工作表的操作中，在单击【导入文本数据】按钮时，为什么总弹出错误提示信息，提示用户无法找到要导入文件的路径？

出现这种情况时，首先查看导入文件的存放路径是否输入正确，在输入文件存放路径时，应根据实际情况相应地调整存放路径的语句：Filename = “D:\Employee\员工信息.txt”。快速查看文件存放路径的方法是：右击“数据.txt”文本文件，在弹出的快捷菜单中选择【属性】菜单项，打开【数据 属性】对话框，在【常规】选项卡中的【位置】项中用户可以看到该文本文件所在位置的完整路径，然后使用这个完整路径替换上述代码中的数据路径即可。

技巧 2：删除多余保护的工作表

在使用 Excel VBA 实现对多个工作表保护操作时，误单击了组合框中的选项，将不需要保护的工作表添加到列表框中，如何才能将不需要保护的工作表从列表中删除掉？

要在列表中删除不需要保护的工作表，可以设置列表框双击事件，即在代码窗口中输入 ListBox1_DblClick 事件处理程序的代码：

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
ListBox1.RemoveItem ListBox1.ListIndex
  '设置当双击列表框的项目时系统自动删除当前项目
End Sub
```

输入这段代码后，再次双击列表框中显示的内容时，系统都会自动将其删除。

第 16 章　开发员工信息表

员工的信息表中包含了员工的各种重要的个人信息，如员工的姓名、学历、编号、职位等。管理员工信息是企业管理中的重要部分。本章介绍的员工信息管理功能包括设计员工信息表的录入窗体、编写 VBA 程序实现逻辑管理。利用这些技术可以实现让用户通过窗体来录入员工的基本信息，通过本章的学习，读者可以熟练掌握 VBA 中常用控件的应用方法。



16.1 使用窗体录入报表数据

窗体有两种功能，一种是提供展示平台，另一种就是实现用户的业务逻辑的交互。本章设计的员工信息管理功能将使用窗体来收集员工的基本信息。读者在添加完各种窗体控件之后，可以设置控件自身的各种属性、事件和方法，并为这些控件编写 VBA 控制代码。本节将介绍创建窗体和实现窗体与工作簿之间交换数据的操作过程。

16.1.1 创建用户窗体

创建用户窗体是实现员工信息管理的第一步，创建好窗体之后，用户才能通过窗体编写 VBA 代码来管理员工的个人基本信息。下面首先来介绍创建窗体的过程。

01 创建〝公司员工个人信息表〞，表中的内容如图 16-1 所示。创建完成之后切换到【开发工具】选项卡，单击【代码】选项组中的 Visual Basic 按钮，打开 VBA 代码编辑窗口。

02 打开 VBE 环境之后，依次选择【插入】➢【用户窗体】菜单命令，插入一个名为 UserForm1 的用户窗体，然后在窗体的【属性】窗口中修改 Caption 属性值为〝录入员工信息〞，完成之后，该窗体的标题文本将显示为输入的文字，如图 16-2 所示。

03 单击【工具箱】窗口中【框架】按钮，在名称为 UserForm1 的用户窗体中绘制一个框架控件，并修改其【属性】窗口中的 Caption 属性值为〝性别〞，更改该框架的显示标题。

04 在【性别】框架中添加员工性别信息。单击【工具箱】窗口中的【单选按钮】控件，在【性别】框架中绘制两个【单选按钮】控件，并修改单选按钮的 Caption 属性值分别为〝男〞和〝女〞，如图 16-4 所示。

05 接下来按照同样的方法在用户窗体上添加【学历】和【员工职位】框架以及各个框架下相应的单选按钮，添加之后的 UserForm1 窗体如图 16-5 所示。

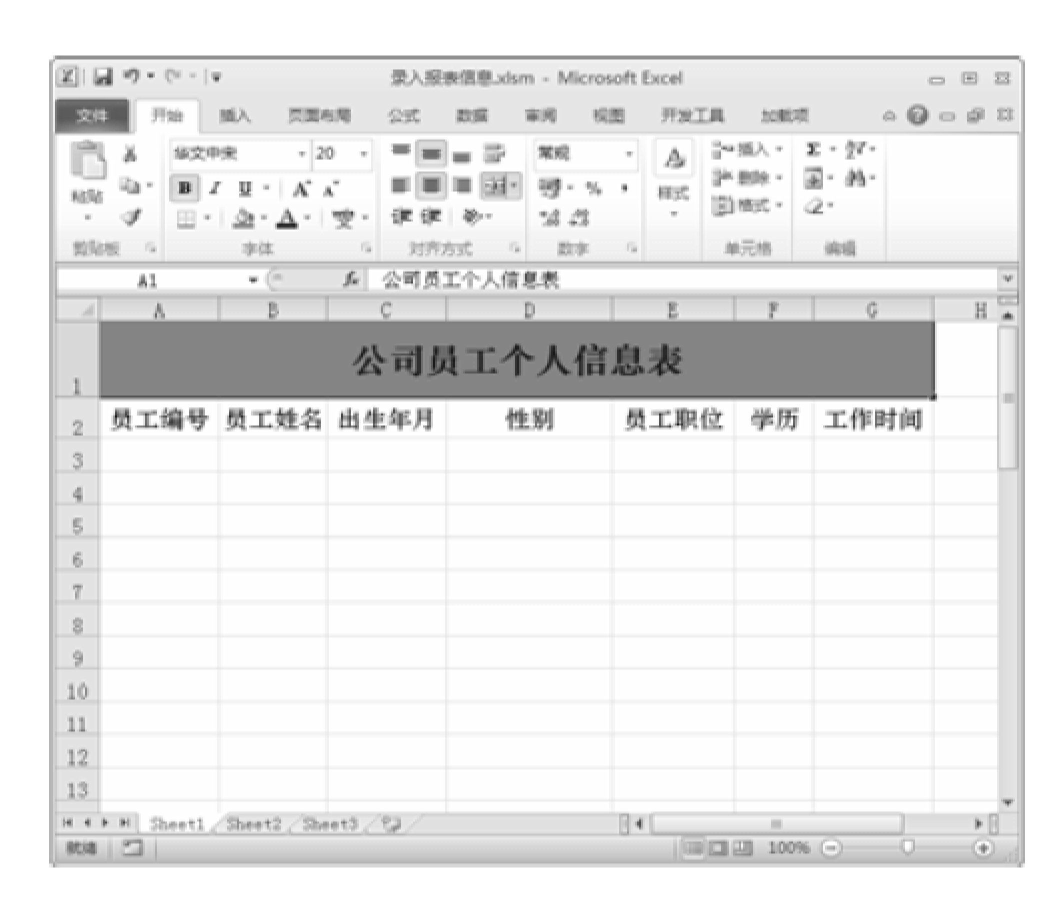


图 16-1　公司员工个人信息表

图 16-2　插入用户窗体

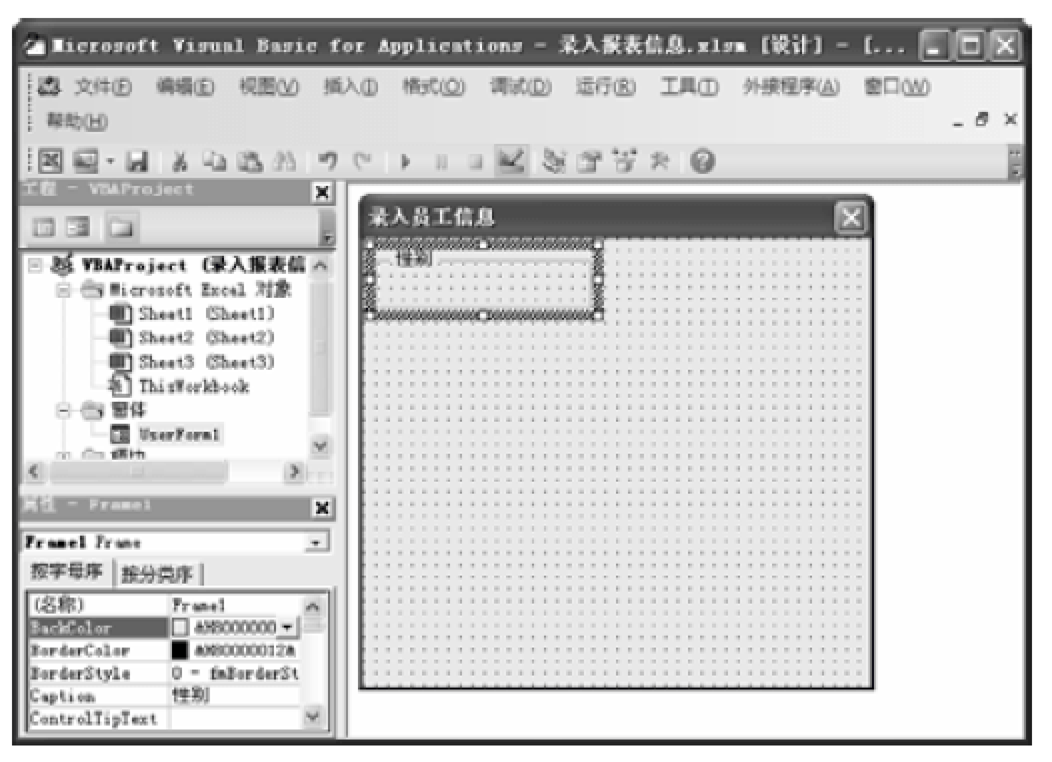


图 16-3　添加框架控件



图 16-4　向“性别”框架中的添加两个选项按钮

【注意】添加【员工职位】框架中的职位单选按钮时，因为选项比较多，所以添加的控件的位置可能不整齐，为了使添加的单选按钮保持整齐，需要对这些控件的位置进行调整。首先全部选中【员工职位】框架中的这些单选按钮（方法是按住 Ctrl 键，然后依次选中这些单选按钮），选中之后，在单选按钮上右击，在弹出的快捷菜单中选择【统一尺寸】➢【两者都相同】菜单命令，这样就可以将选中的所有单选按钮设置为统一的大小，如图 16-6 所示。



图 16-5　添加其他框架及其选项

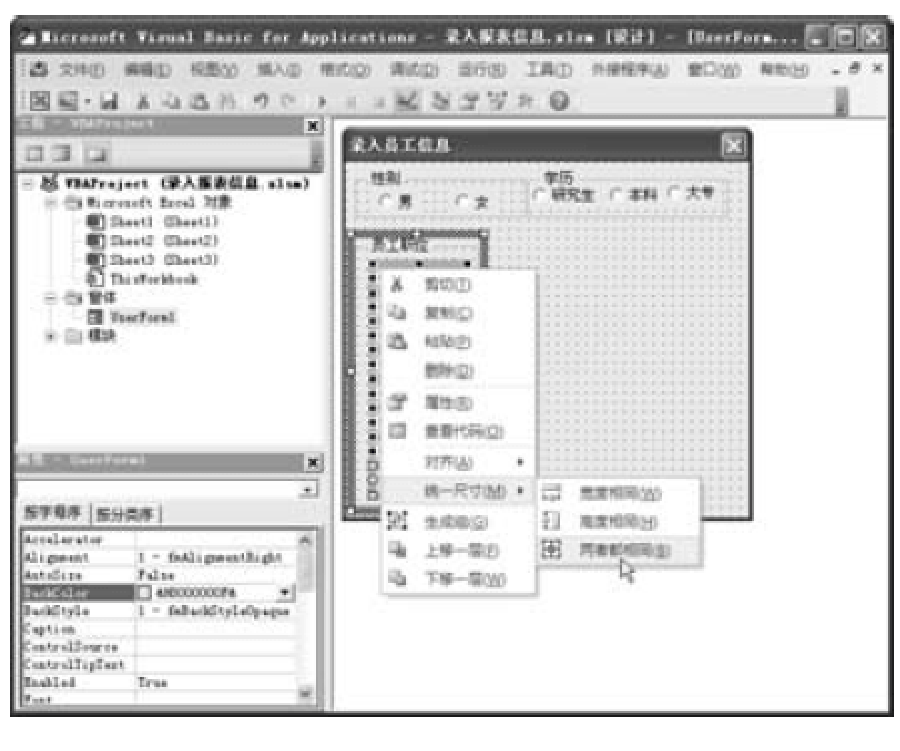


图 16-6　统一控件大小

然后在选中的单选按钮上再次右击，在弹出的快捷菜单中选择【对齐】➤【对齐到网格】菜单命令，将所有选中的单选按钮设置为统一的对齐方式，如图 16-7 所示。

最后使用鼠标拖动控件调整各单选按钮的位置，调整之后的效果如图 16-8 所示。

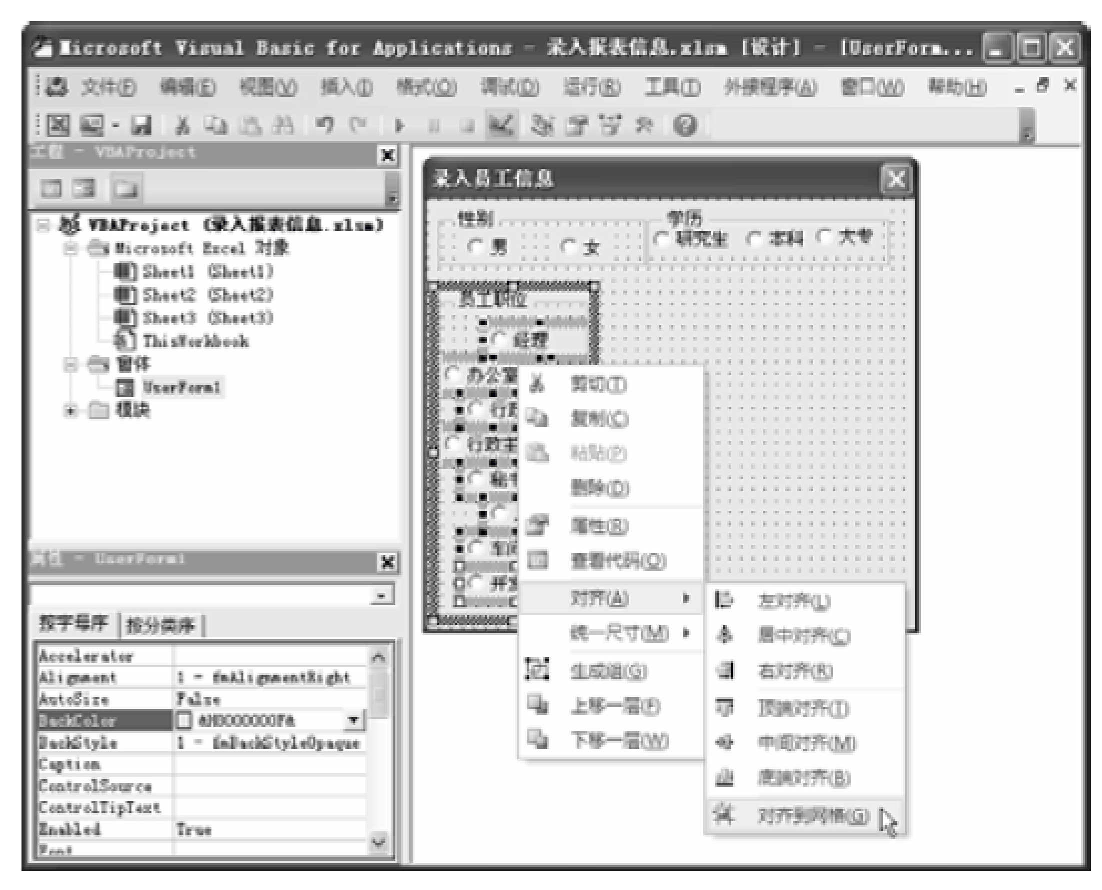


图 16-7 【对齐到网格】命令



图 16-8 调整各单选按钮的位置

06 有些员工信息是不能选择输入的，如员工的姓名、编号等。接下来，添加一些可以让用户手动输入数据的控件。

首先，单击【工具箱】窗口中【标签】按钮，按住鼠标左键在用户窗体中拖动鼠标，绘制一个标签控件，修改其 Caption 属性值为“员工编号”，然后修改 BorderStyle 属性值为下拉列表中的 1-fmBorderStyleSingle 选项，表示该标签显示边框，如图 16-9 所示。

然后，单击【工具箱】窗口中【文本框】按钮，在用户窗体中绘制一个文本框控件，如图 16-10 所示。




图 16-9 添加标签控件



图 16-10 添加文本框控件

按照同样的方法，在用户窗体中继续添加【员工姓名】、【出生年月】和【入职日期】标签以及与其相对应的文本框，如图 16-10 所示。

07 接下来，添加操作确认与取消按钮，单击【工具箱】窗口中【命令按钮】，在用户窗体上绘制两个命令按钮控件，分别修改按钮的 Caption 属性值为"确定"和"取消"，如图 16-11 所示。



图 16-10　添加其他标签以及与其对应的文本框



图 16-11　添加"确定"和"取消"按钮

16.1.2 窗体与工作簿的数据交换

完成用户窗体的创建之后，下面就需要编写 VBA 代码来实现窗体与工作簿中数据的交换过程，操作步骤如下。

01 在 VBE 环境中依次选择【插入】➢【模块】菜单项，插入模块 1，在该模块代码编辑窗口中添加下面的程序。

```
Option Explicit

Public em_num As String    '定义用于记录"员工编号"的变量
Public em_name As String   '定义用于记录"员工姓名"的变量
Public em_sex As String    '定义用于记录"性别"的变量
Public em_born As String    '定义用于记录"出生年月"的变量
Public em_degree As String  '定义用于记录"学历"的变量
Public em_year As String    '定义用于记录"工作时间"的变量
Public em_job As String    '定义用于记录"员工职位"的变量
Sub InitEmVar()    '初始化全局变量
    em num = ""
    em name = ""
    em sex = ""
    em_born = ""
    em_degree = ""
    em_year = ""
    em_job = ""
End Sub
Sub getOneRecord()     '录入功能实现
    Dim MsgBoxRst As VbMsgBoxResult
    Dim count As Integer
```




```
        If UserForm1.TextBox1.Text = "" Or UserForm1.TextBox2.Text = "" Or
          UserForm1.TextBox3.Text = "" Or UserForm1.TextBox4.Text = "" Or
          em_sex = "" Or em_degree = "" Or em_job = "" Then
          MsgBoxRst = MsgBox("请输入完整的信息！", vbOKOnly & vbInformation, "系统
提示")
          Exit Sub
        End If
        count = 0     '为变量赋初值
        Range("A2").Select
        '判断是否输入重复的员工编号
        Do Until Selection.Offset(count, 0).Value = ""
            If UserForm1.TextBox1.Text = Selection.Offset(count, 0).Value Then
                MsgBoxRst=MsgBox("员工编号重复,请重新输入!",vbOKOnly&vbInformation,
"提示")
                Exit Sub
            End If
            count = count + 1
        Loop

        With UserForm1
            em_num = .TextBox1.Text
            em_name = .TextBox2.Text
            em_born = .TextBox3.Text
            em_year = .TextBox4.Text
        End With

        count = 0
        Range("A2").Select
        Do Until Selection.Offset(count, 0).Value = ""
            count = count + 1
        Loop
        '向工作表单中插入一条员工个人信息
        Selection.Offset(count, 0).Value = em_num
        Selection.Offset(count, 1).Value = em_name
        Selection.Offset(count, 2).Value = em_born
        Selection.Offset(count, 3).Value = em_sex
        Selection.Offset(count, 4).Value = em_job
        Selection.Offset(count, 5).Value = em_degree
        Selection.Offset(count, 6).Value = em_year
        Call InitEmVar
        If MsgBox("是否继续录入?", vbYesNo, "继续录入") = vbNo Then
            UserForm1.Hide
        Else
            With UserForm1
                .TextBox1.Text = ""
                .TextBox2.Text = ""
                .TextBox3.Text = ""
                .TextBox4.Text = ""
            End With
        End If
```

```
End Sub
```

该段代码开始定义了几个保存员工个人信息的全局变量，然后利用 InitEmVar 过程对这些全局变量初始化。getOneRecord 过程的作用是将窗体上各种控件的值转换成用户可以理解的员工信息并保存到 Excel 工作表相应的单元格中。

02 双击用户窗体中【确定】按钮，进入该按钮的 Click 事件代码编辑窗口，在 Click 事件中调用 getOneRecord 过程，添加代码如下。

```
Private Sub CommandButton1_Click()
    getOneRecord
End Sub
```

这段代码表示当单击【确定】按钮时，系统将调用 getOneRecord 过程插入一条员工信息。

03 双击用户窗体中的【取消】按钮，进入该按钮的 Click 事件代码编辑窗口，在 Click 事件中调用 InitEmVar 过程并隐藏添加员工信息的用户窗体 UserForm1，添加代码如下。

```
Private Sub CommandButton2_Click()
Call InitEmVar
UserForm1.Hide
End Sub
```

这里 UserForm1 的 Hide 方法表示隐藏该窗体。

04 接下来介绍如何处理 3 个框架中的各个单选按钮。在这里分别双击这些单选按钮，进入其 Click 事件的代码编辑窗口，然后分别在其代码窗口中添加与之相对应的代码，每个单选按钮的 Click 事件都将保存员工的一项个人信息，具体代码如下：

```
Private Sub OptionButton1 Click()
    em sex = "男"
End Sub

Private Sub OptionButton2 Click()
    em sex = "女"
End Sub

Private Sub OptionButton3 Click()
    em degree = "研究生"
End Sub


Private Sub OptionButton4 Click()
    em degree = "本科"
End Sub

Private Sub OptionButton5 Click()
    em degree = "大专"
End Sub

Private Sub OptionButton6 Click()
    em_job = "经理"
```

```
End Sub

Private Sub OptionButton7 Click()
    em job = "办公室主任"
End Sub

Private Sub OptionButton8 Click()
    em job = "行政助理"
End Sub

Private Sub OptionButton9 Click()
    em job = "行政主管"
End Sub

Private Sub OptionButton10 Click()
    em job = "秘书"
End Sub

Private Sub OptionButton11 Click()
    em job = "后勤主管"
End Sub

Private Sub OptionButton12 Click()
    em job = "车间工人"
End Sub

Private Sub OptionButton13 Click()
    em job = "开发人员"
End Sub
```

现在已经完成了窗体数据与工作簿中数据交换的 VBA 代码的编写。下面将介绍如何在 Excel 表单中调用用户窗体。

05 返回到 Excel 中 Sheet1 工作表，切换到【开发工具】选项卡，在【控件】选项组的【插入】下拉列表中单击【命令按钮】按钮 （ActiveX 控件），然后拖动鼠标在工作表的适当位置绘制一个命令按钮控件，并修改“名称”和 Caption 两个属性的值，两个值都为“输入个人信息”，如图 16-12 所示。

06 单击【设计模式】按钮，然后双击【输入个人信息】按钮控件，进入该按钮的 Click 事件代码编辑窗口，在其中添加调用用户窗体的 VBA 代码，如图 16-13 所示。

```
Private Sub 输入个人信息_Click()
    With UserForm1
        .TextBox1.Text = ""
        .TextBox2.Text = ""
        .TextBox3.Text = ""
        .TextBox4.Text = ""
        .Show
    End With
    Call InitEmVar    '调用 InitEmVar 过程
End Sub
```

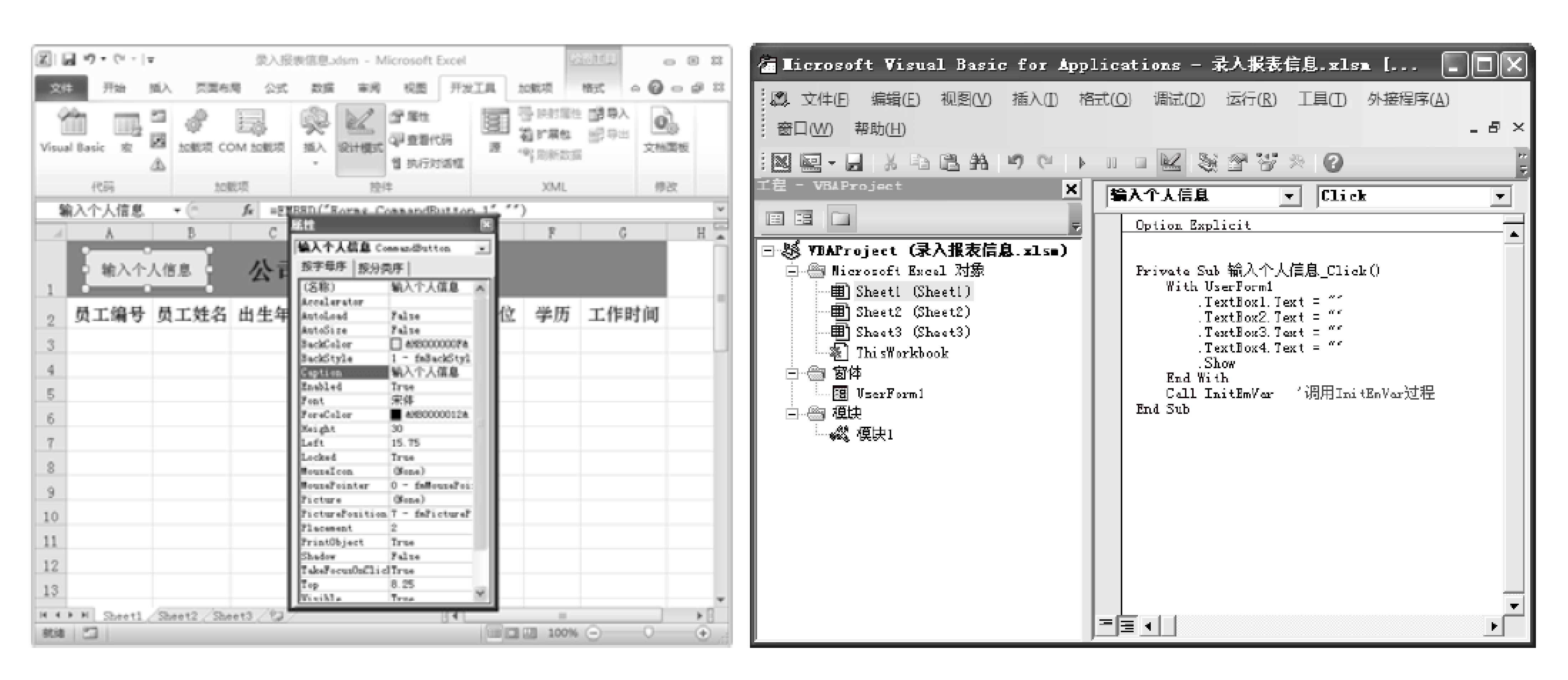


图 16-12　插入按钮控件　　　　图 16-13　编写按钮 Click 事件代码

07 返回 Excel 表单，再次单击【设计模式】按钮退出按钮的设计状态，完成用户窗体调用代码的编写。此时可以单击工作表中的【输入个人信息】按钮，打开【录入员工信息】对话框，结果如图 16-14 所示，至此已经完成所有代码的编写了。

08 现在，用户已经可以使用编写好的程序来录入员工的个人信息了，在打开的对话框中选中单选按钮或者在文本框中输入员工的各项个人信息，如图 16-15 所示。



图 16-14　【录入员工信息】对话框　　　　图 16-15　输入员工数据

输入完毕，单击【确定】按钮，即可将各项员工信息导入到 Sheet1 工作表中。信息导入成功之后，会弹出一个提示窗口，提示用户是否要继续录入数据，如图 16-16 所示，单击【是】按钮将返回【录入员工信息】对话框，单击【否】按钮将退出【录入员工信息】对话框并返回 Excel 表格。返回之后，便可在工作表中看到录入的员工信息，如图 16-17 所示。

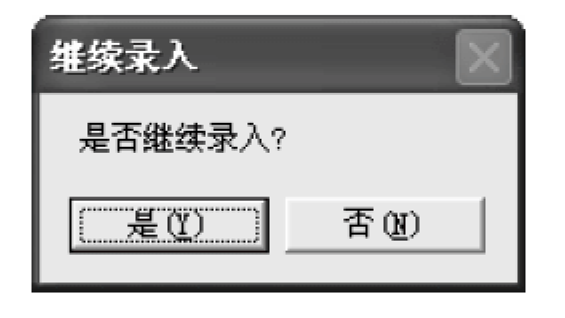


图 16-16 【继续录入】对话框

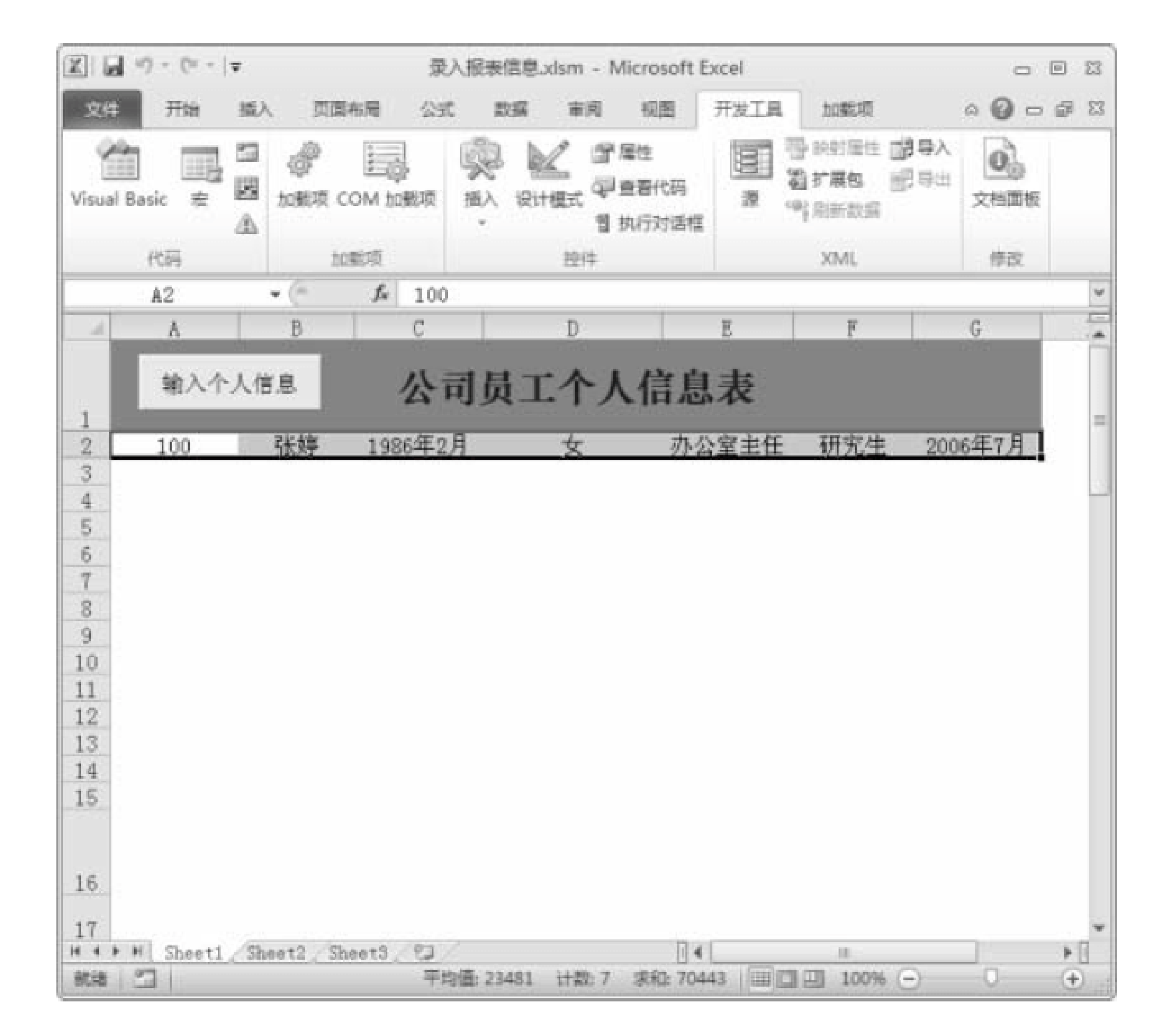


图 16-17 显示录入的员工信息

【提示】在程序代码中，对可能出现的错误进行了一些处理。例如，如果在【录入员工信息】对话框中输入了不完整的信息或没有输入任何信息，单击【确定】按钮后系统将弹出消息提示对话框，提示用户“请输入完整的信息”，如图 16-18 所示。如果用户输入的“员工编号”已经存在，当单击对话框中的【确定】按钮时，系统会弹出【提示】消息框，提示用户“员工编号重复，请重新输入”，如图 16-19 所示。

09 用户可按照前面的步骤利用【录入员工信息】对话框快速方便地录入报表数据，如图 16-20 所示。



图 16-18 系统提示信息

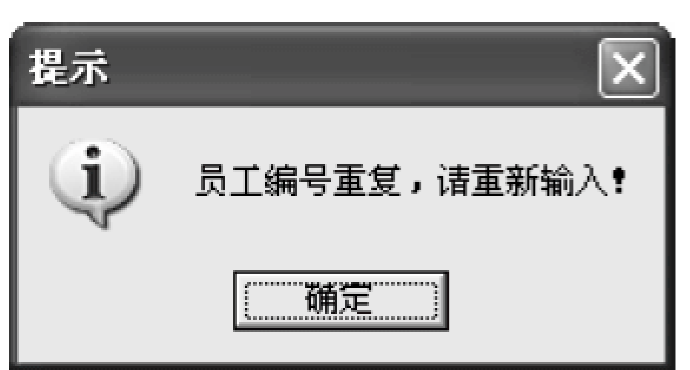


图 16-19 提示编号重复



图 16-20 利用用户窗体录入报表数据



16.2 管理员工数据报表

现在已经有了员工的基本信息，用户可以使用这些信息来制作员工卡、生成员工工资条等。本节将介绍如何在 Excel 中利用 VBA 代码实现这些功能。

16.2.1 员工卡的制作

员工卡包含了员工的基本信息。首先来看如何利用 VBA 代码制作员工卡，具体操作步骤如下。

01 使用 16.1 节介绍的方法创建一张员工信息报表并将其打开，或者直接将"录入报表信息.xlsm"文件拷贝一份，并将其重命名为"员工卡"，然后打开该文件。首先在工作表的上方插入适当的空白行来存放员工卡中的信息，并取消工作表中网格线的显示效果。打开 VBE 环境，依次选择【插入】➤【模块】菜单命令，即可插入模块 2，在该模块窗口中输入如下代码：

```
Public Sub 插入多行()
    Dim idx  As Integer
    For idx  = 1 To 15  '插入 15 行
       Rows(1).Insert
    Next idx
    ActiveWindow.DisplayGridlines = False      '取消工作表中网格线的显示
End Sub
```

输入完毕后，依次选择【运行】➤【运行子过程/用户窗体】菜单命令，返回到 Excel 表单，结果如图 16-21 所示。



图 16-21　插入空白行效果

02 选中单元格区域 A1:E13 作为保存员工卡信息的区域，单击【开始】选项卡中的【颜色

填充】按钮右侧的下三角按钮，在弹出的下拉列表中选择想要的颜色，如图 16-22 所示。



图 16-22 填充区域颜色

03 设置标签控件属性。首先切换到【开发工具】选项卡，在【控件】选项组中的【插入】下拉列表中单击【标签】按钮A（ActiveX 控件），当鼠标指针变为"+"形状时，在工作表中适当位置按住鼠标左键进行拖动，拖动至合适大小后释放鼠标左键添加一个标签控件。

右击该控件，选择弹出菜单中的【属性】命令，打开控件的【属性】对话框，切换到【按字母序】选项卡，修改 Caption 属性值为“员工卡”。

单击 Font 属性选项后面的【选择】按钮...，打开【字体】对话框，从中设置字体、字形和大小，如图 16-23 所示。

单击【确定】按钮，保存字体的设置并返回【属性】对话框，单击 ForeColor 选项后面的下三角按钮▼，在其下拉列表中切换到【调色板】选项卡，选择适当的颜色选项，如图 16-24 所示。

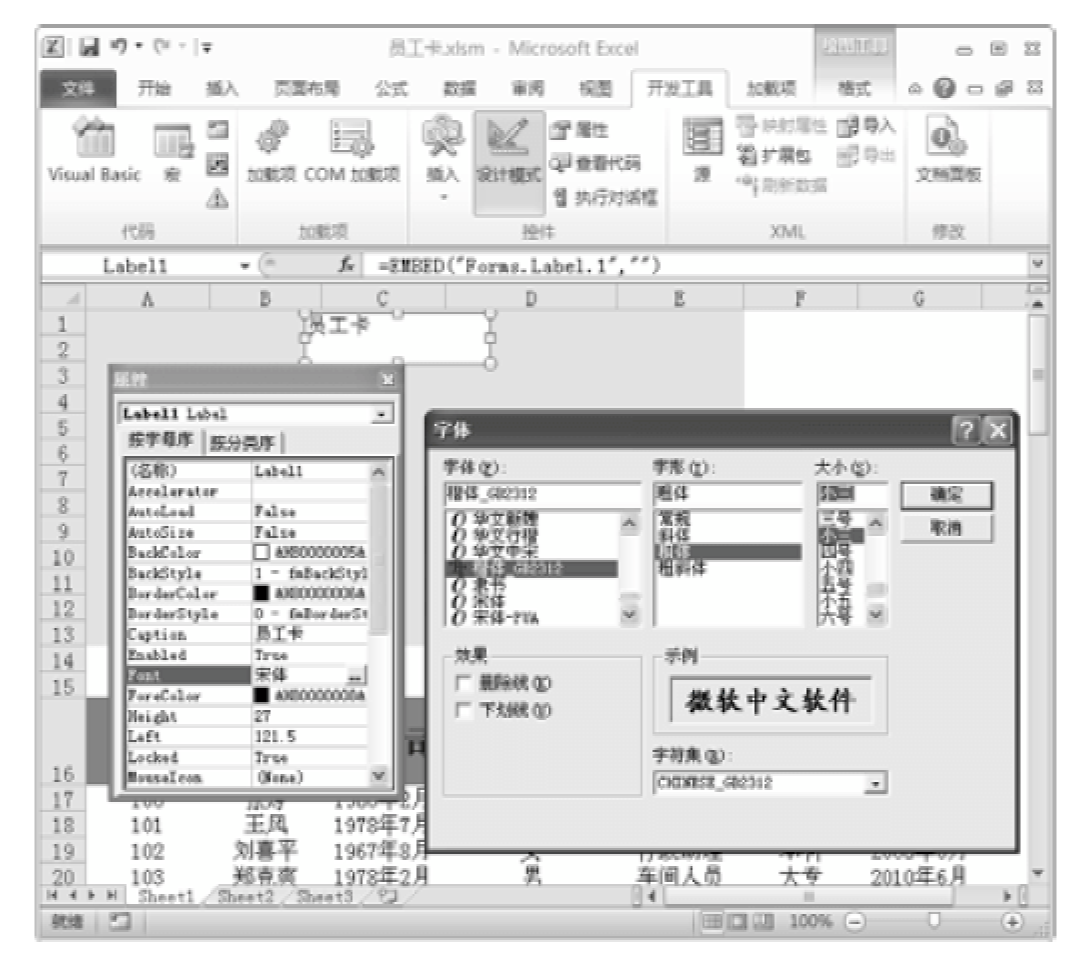


图 16-23 设置标题字体



图 16-24 设置标题颜色

单击 Shadow 属性选项后面的下三角按钮，选择下拉列表中的 True 选项，然后单击 TextAling 属性后面的下三角按钮，在下拉列表中选择 2-fmTextAlignCenter 选项，设置控件的阴影和居中效果，如图 16-25 所示。

04 接下来按照同样的方法添加其他控件，首先添加名称为“编号”的 Label 控件，然后单击控件列表中的【文本框】按钮，当鼠标指针变为“+”形状时，在工作适当的位置按住鼠标左键不放，拖至适当大小后释放鼠标左键添加一个文本框控件，如图 16-26 所示。





图 16-25　设置控件的阴影和居中效果　　　　图 16-26　添加文本框控件

05 按照同样的方法添加其他员工卡信息控件；然后添加一个保存员工头像的控件，单击控件列表中的【图像】按钮（ActiveX 控件），当鼠标指针变为“+”形状时，在工作表中按住鼠标左键不放，拖至适当大小后释放鼠标左键即可添加一个图像控件，如图 16-27 所示。

06 接下来设置姓名组合框。首先选中 B18:B29 单元格区域，然后切换到【公式】选项卡，在【定义的名称】选项组中选择【定义名称】➤【定义名称】选项，打开【新建名称】对话框，在【名称】文本框中输入“员工姓名”，然后单击【确定】按钮即可完成对所选区域名称的定义，如图 16-28 所示。





图 16-27　添加一个图像控件　　　　图 16-28　【新建名称】对话框

07 添加组合框和命令按钮控件。在该工作表中添加一个保存员工姓名的组合框，用户可以从中选择要显示信息的员工姓名。首先，切换到【开发工具】选项卡，单击控件列表中的【组合框】按钮 （ActiveX 控件），在工作表的适当位置添加一个组合框控件。控件添加完成之后，设置其 ListFillRange 属性值为"员工姓名"，如图 16-29 所示。

单击控件列表中的【命令按钮】按钮 （ActiveX 控件），在工作表中添加命令按钮控件，并修改"名称"和 Caption 两个属性的值为"生成员工卡信息"，如图 16-30 所示。

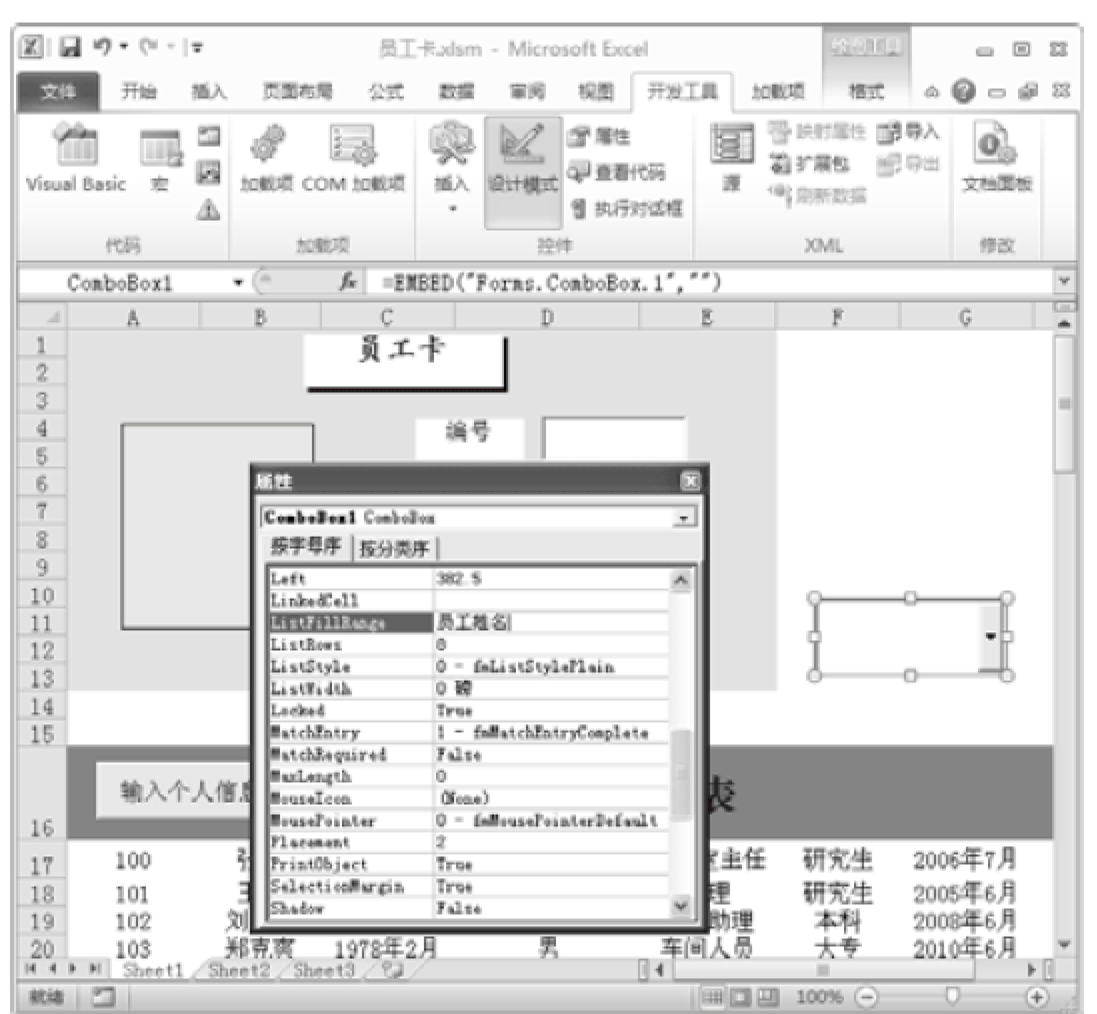


图 16-29 添加保存员工姓名的组合框



图 16-30 添加显示信息的按钮控件

08 在设计模式下双击【生成员工卡信息】按钮，进入该按钮的 Click 事件代码编辑窗口，输入 VBA 代码如下。

```
Private Sub 生成员工卡信息_Click()
    Dim rag As Range '定义区域变量
    Dim row As Integer
    Dim w As Integer
    Dim h As Integer
    row = 0 '为变量赋初值
    w = 0
    h = 0
    '根据组合框中显示的姓名，在单元格区域中查找该员工并返回该员工在工作表中的具体位置
    For Each rag In Range("B17:B29")
        If rag.Value = ComboBox1.Text Then
            row = rag.Row
        End If
    Next
    Dim myPath As String
    myPath = Application.ThisWorkbook.Path

    '将员工的对应信息返回到员工卡区域中
    If row > 0 Then
```

```
            TextBox1.Text = Cells(row, 1).Value
            TextBox2.Text = Cells(row, 2).Value
            TextBox3.Text = Cells(row, 3).Value
            TextBox4.Text = Cells(row, 4).Value

            With Image1 '调用照片并显示在图像控件中
                w = .Width
                h = .Height
                .Visible = False
                .Picture = LoadPicture(myPath & "\员工头像\" & Cells(row, 8).Value)
                .AutoSize = True
                .Width = w
                .Height = h
                .Visible = True
            End With
        Else
            MsgBox "该员工姓名不存在，请重新核实！", vbOKOnly & vbInformation, "提示"
        End If
    End Sub
```

【提示】这里，在调用照片并显示在图像控件中时，使用的是图片的相对路径，下面的一行可以实现获取当前表格所在的系统路径：

```
    myPath = Application.ThisWorkbook.Path
```

用户可根据实际情况重新输入照片的存放路径。

09 现在已经完成了所有代码的编写和控件的设置，可以生成员工卡信息了。再次单击 VBE 环境工具栏中的【设计模式】按钮，退出按钮的编辑状态，返回到工作表中，在组合框的下拉列表中选择员工姓名后，单击【生成员工卡信息】按钮，即可在员工卡中显示该员工的信息，如图 16-31 所示。

用户也可以在组合框中直接输入员工姓名，但是如果输入的员工姓名不存在，例如输入“张大万”，当单击【显示信息】按钮时，系统则会弹出该员工姓名不存在的提示信息，如图 16-32 所示。

【提示】如果想使整体效果更加美观，可以将员工信息报表数据区域隐藏，方法是选中工作表中的第 17~29 行，然后在行标处单击鼠标右键，在弹出的快捷菜单中选择【隐藏】选项，即可隐藏员工信息报表数据，如图 16-33 所示。

如果用户想查看工作表中记录的员工信息报表，则需要取消对单元格的隐藏。即选中第 17~29 行，然后在行标上单击鼠标右键，在弹出的快捷菜单中选择【取消隐藏】菜单项，即可取消隐藏，如图 16-34 所示。





图 16-31　显示员工信息



图 16-32　系统提示信息

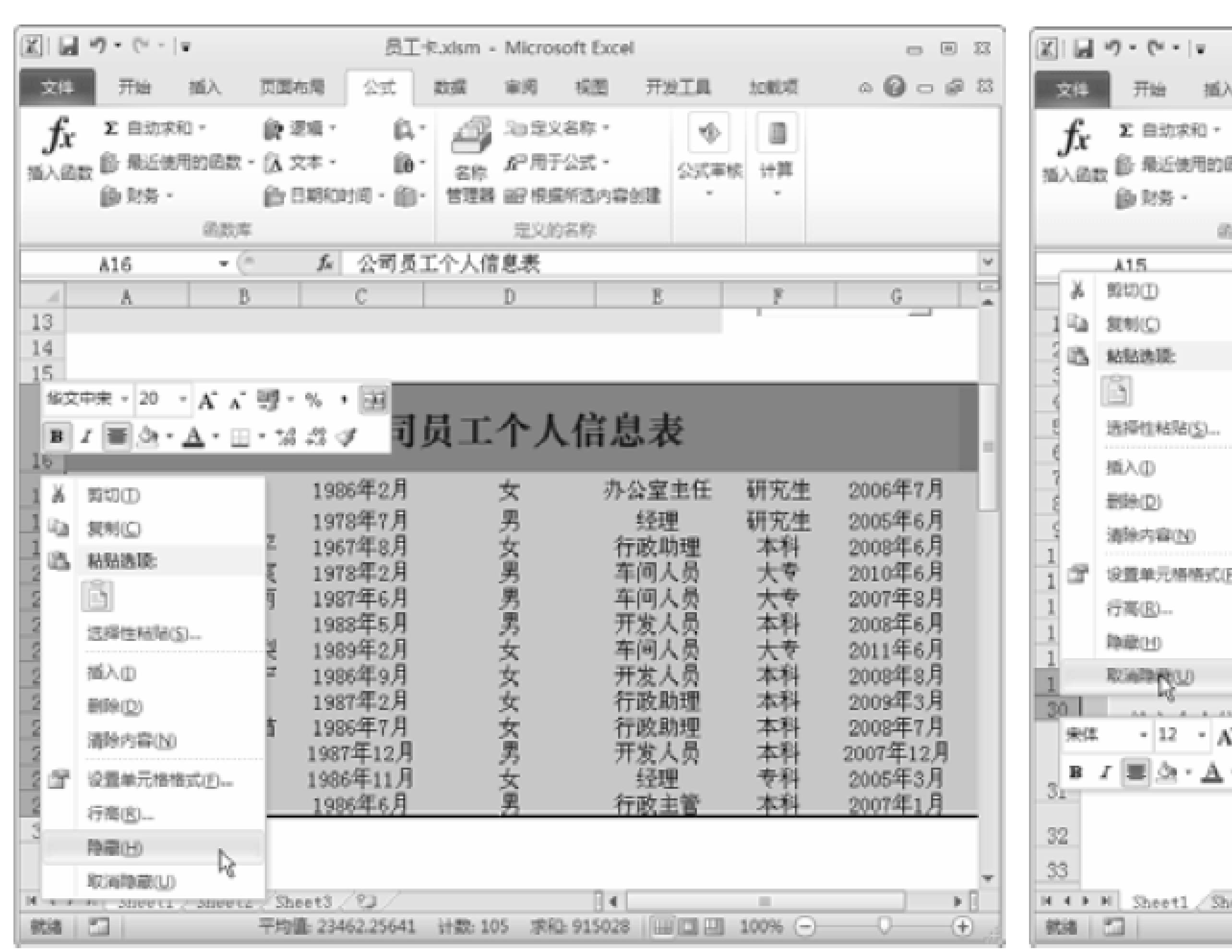


图 16-33　隐藏单元格



图 16-34　取消隐藏单元格

使用下面的 VBA 代码也可以实现单元格的隐藏操作。隐藏单元格的 VBA 语句如下：

```
Rows("16:29").Select
Selection.EntireRow.Hidden = True
```

取消单元格隐藏操作的 VBA 语句如下：

```
Rows("16:29").Select
Selection.EntireRow.Hidden = False
```

员工卡的制作方法到这里就介绍完了。

该段程序使用了一个系统函数 Format。Format 函数的功能是返回格式化的字符串数据 Variant（String）值，其中含有一个表达式，它是根据格式表达式中的指令来格式化的。Format 函数的语法格式为：

```
Format(expression[,format[,firstdayofweek[,firstweekofyear]]])
```

其各参数的具体意义如表 16-1 所示。

表 16-1　Format 参数的具体意义

参数	说明
expression	代表任何有效的表达式，必要参数
format	代表有效的命名表达式或用户自定义格式表达式，可选参数
firstdayofweek	常数，表示一星期的第一天，可选参数
firstweekofyear	常数，表示一年的第一周，可选参数

其中关于参数 firstdayofweek 的设置如表 16-2 所示。

表 16-2　参数 firstdayofweek 的设置

常数	值	说明
vbUseSystem	0	使用 NLS API 设置
vbSunday	1	星期日（默认）
vbMonday	2	星期一
vbTuesday	3	星期二
vbWednesday	4	星期三
vbThursday	5	星期四
vbFriday	6	星期五
vbSaturday	7	星期六

关于参数 firstweekofyear 的设置如表 16-3 所示。

表 16-3　参数 firstweekofyear 的设置

常数	值	说明
vbUseSystem	0	使用 NLS API 设置
vbFirstJan1	1	从包含 1 月 1 日那一周开始（默认）
vbFirstFourDays	2	从本年第一周开始，而此周至少有 4 天在本年中
vbFirstFullWeek	3	从本年第一周开始，而此周完全在本年中

按照“hh:mm:ss”格式显示时间 ，Format 格式参数如下：

```
format(time,"hh:mm:ss")
```

按照“yyyy 年 m 月 d 日”格式显示时间系统，Format 格式参数如下：

```
format(date,"yyyy 年 m 月 d 日")
```

想要设置保留小数点后 3 位数字且为货币形式的格式，Format 格式参数如下：

```
format("0.12345","$0.000")
```

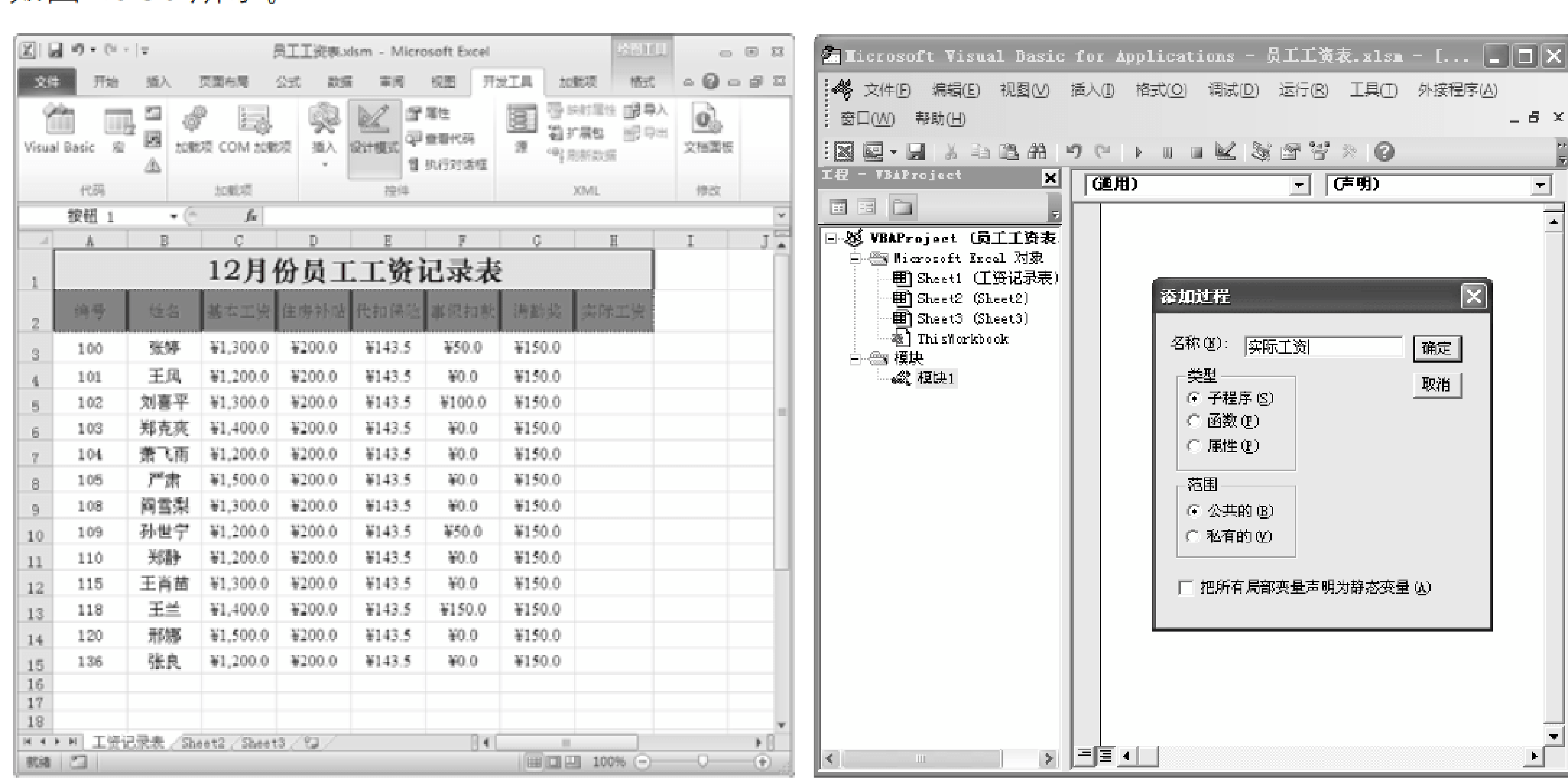
其返回值为“$0.123”。

16.2.2 工资条的生成

员工工资条包含了每位员工的姓名和对应的各项工资信息。下面将介绍如何利用 VBA 快速生成员工工资条，具体的操作步骤如下：

01 新建员工工资表工作簿，将 Sheet1 工作表重命名为“工资记录表”，然后在该工作表中输入员工各项工资信息，如图 16-35 所示。

02 添加“实际工资”子过程。首先切换到【开发工具】选项卡，从中单击【代码】选项组中的 Visual Basic 按钮，打开 VBA 代码编辑窗口，从中选择【插入】➤【模块】菜单项，插入一个模块 1，然后在该模块窗口中，选择【插入】➤【过程】菜单项，打开【添加过程】对话框，设置子过程的属性如下：【名称】为“实际工资”，【类型】为“子程序”，【范围】为“公共的”，如图 16-36 所示。



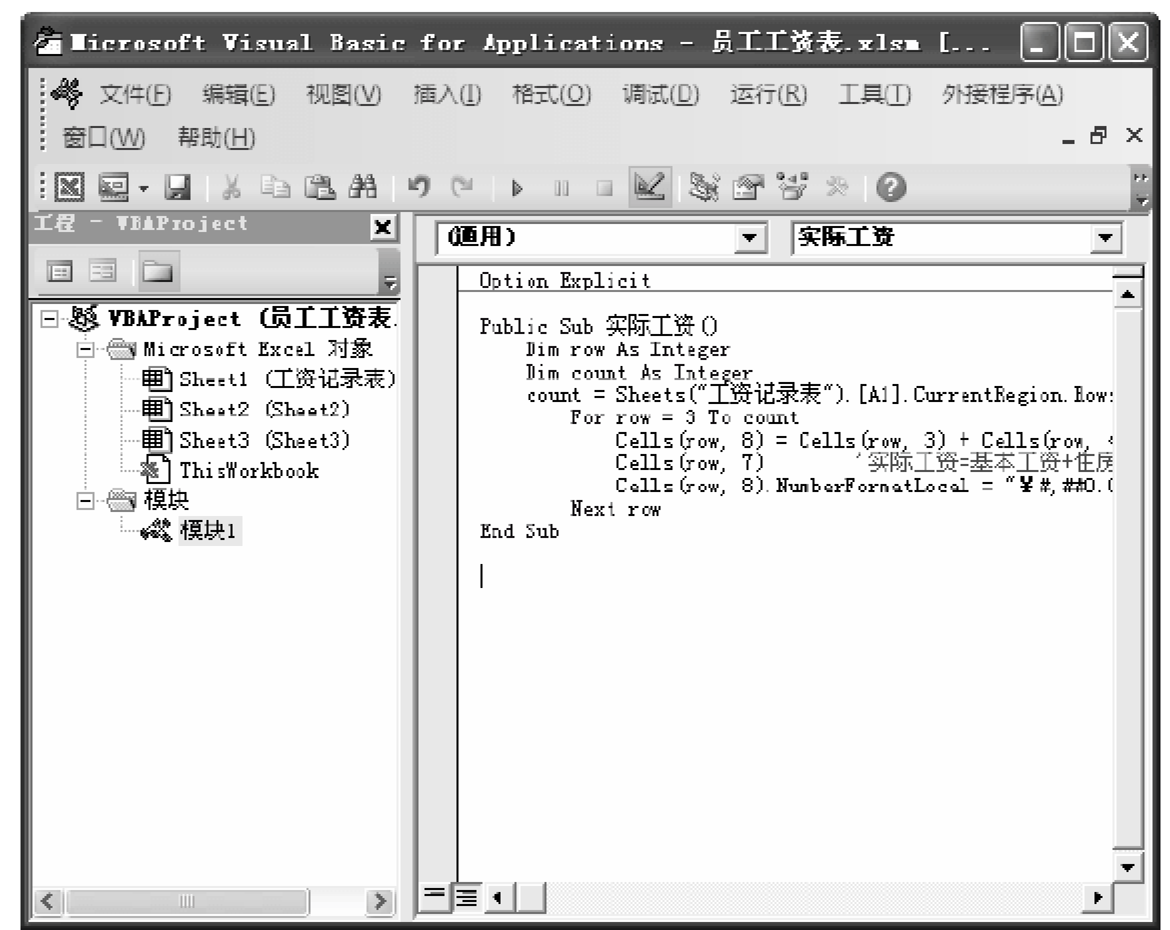
12月份员工工资记录表							
编号	姓名	基本工资	住房补贴	代扣保险	事假扣款	满勤奖	实际工资
100	张婷	¥1,300.0	¥200.0	¥143.5	¥50.0	¥150.0	
101	王凤	¥1,200.0	¥200.0	¥143.5	¥0.0	¥150.0	
102	刘喜平	¥1,300.0	¥200.0	¥143.5	¥100.0	¥150.0	
103	郑克爽	¥1,400.0	¥200.0	¥143.5	¥0.0	¥150.0	
104	萧飞雨	¥1,200.0	¥200.0	¥143.5	¥0.0	¥150.0	
105	严肃	¥1,500.0	¥200.0	¥143.5	¥0.0	¥150.0	
108	阎雪梨	¥1,300.0	¥200.0	¥143.5	¥0.0	¥150.0	
109	孙世宁	¥1,200.0	¥200.0	¥143.5	¥50.0	¥150.0	
110	郑静	¥1,200.0	¥200.0	¥143.5	¥0.0	¥150.0	
115	王肖茵	¥1,300.0	¥200.0	¥143.5	¥0.0	¥150.0	
118	王兰	¥1,400.0	¥200.0	¥143.5	¥150.0	¥150.0	
120	邢娜	¥1,500.0	¥200.0	¥143.5	¥0.0	¥150.0	
136	张良	¥1,200.0	¥200.0	¥143.5	¥0.0	¥150.0	

图 16-35　员工工资表　　　　图 16-36　【添加过程】对话框

单击【确定】按钮即可创建名为“实际工资”的公有子过程。在其中输入该过程中要执行的 VBA 代码，结果如图 16-37 所示。

```
Public Sub 实际工资()
    Dim row As Integer
    Dim count As Integer
    count = Sheets("工资记录表").[A1].CurrentRegion.Rows.count 'count 变量保存工作表中总的记录数
        For row = 3 To count
            Cells(row, 8) =Cells(row, 3) +Cells(row, 4) +Cells(row, 7) -Cells(row, 5) -Cells(row, 6)         '实际工资=基本工资+住房补贴+满勤奖-事假扣款-代扣保险
            Cells(row, 8).NumberFormatLocal = "￥#,##0.0;￥-#,##0.0" '设置货币显'示格式
```

```
        Next row
    End Sub
```



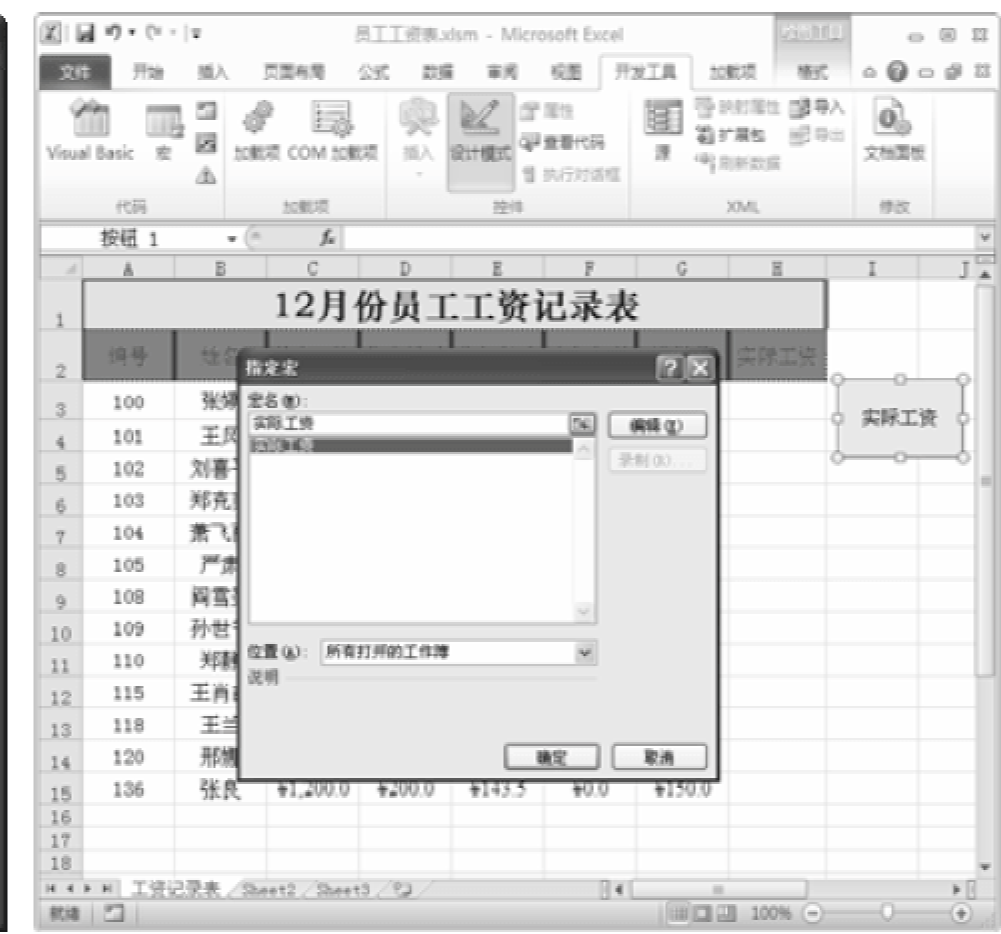


图 16-37　计算实发工资程序代码　　　　图 16-38　【指定宏】对话框

03 在 Excel 表单中添加执行"实际工资"子过程的执行按钮，切换到【开发工具】选项卡，单击【控件】选项组中的【插入】下拉列表中的【按钮（窗体控件）】按钮，当鼠标指针变为"+"形状时，在工作表中按住鼠标左键进行拖动，拖动至合适大小后释放鼠标左键；将按钮名称更改为"实际工资"，单击工作表中其他任意位置退出该按钮的编辑状态，打开【指定宏】对话框，在【宏名】列表框中选择"实际工资"，单击【确定】按钮，完成宏的指定操作，如图 16-38 所示。

将鼠标指针移到该按钮上，当指针形状变为时单击【实际工资】按钮，即可完成实际工资的计算，如图 16-39 所示。

04 接下来开始介绍生成员工工资条的程序设计过程。首先在模块 1 中继续添加子过程"生成工资条"，输入 VBA 代码如下。

```
    Public Sub 生成工资条()
        Dim i As Integer, row As Integer, col As Integer
        Dim j As Integer
        Dim sht As Object
        Dim wsExist As Boolean '判断是否存在指定的工作表
        wsExist = False
        row = Sheets("工资记录表").[A2].CurrentRegion.Rows.count
        col = Sheets("工资记录表").[A2].CurrentRegion.Columns.count
        Range(Cells(2, 1),Cells(row,col)).Copy '从“工资记录表”表单中复制员工工资数据

        For Each sht In Worksheets              '判断是否存在“生成工资条”工作表
            If sht.Name = "工资条表单" Then
                sht.Activate
                wsExist = True
                sht.cleae
                Exit For
            End If
```

```
        Next sht
        'wsExist 值为 False 表示不存在名称为“工资条表单”的表单
        If wsExist = False Then
            Sheets.Add After:=Sheets("工资记录表")'在工作表“工资记录表”的后面插入新工作表
            ActiveSheet.Name = "工资条表单"        '对工作表重命名
            ActiveWorkbook.Sheets("工资条表单").Tab.ColorIndex = 43 '设置新工作表的
标'签颜色为“浅绿”
        End If
        ActiveSheet.Cells(1, 1).Select '选中活动工作表中的第 1 个单元格
        ActiveSheet.Paste   '将“工资记录表”表单中员工工资数据复制到“工资条表单”表单
        Selection.PasteSpecial Paste:=xlPasteColumnWidths, Operation:=xlNone, _
        SkipBlanks:=False, Transpose:=False


        For row = 14 To 3 Step -1 '为每一条员工工资记录添加一个空白行
            Rows(row).Select
            Selection.Insert Shift:=xlDown
            With Selection.Font '设置插入空白行的字体格式
                .Name = "黑体"
                .ColorIndex = 41
                .Size = 12
            End With
            Selection.HorizontalAlignment = xlCenter '设置水平居中
        Next row

        '将工资的列标题赋值给新插入的空白行
        For row = 1 To 26             '工作表中共有 13 行记录数据，生成工资条后将变为 26
行
            If Cells(row, 1) = "" Then    '判断第 1 列内的单元格是否为空
                 For col = 1 To 8       '工作表中共有 8 列数据
                    Cells(row, col) = Cells(1, col)   '将第一行中的标题分别复制到每一个
'员工工资条记录的空白行中
                     Debug.Print Cells(1, col)
                Next col
            End If
        Next row

         MsgBox "成功生成员工工资条！"

    End Sub
```

05 返回“工资记录表”表单，添加执行“生成工资条”过程的按钮控件。在【开发工具】选项卡中，单击【控件】选项组中的【插入】下拉列表中的【按钮（窗体控件）】按钮 ，当鼠标指针变为“+”形状时，在工作表中按住鼠标左键进行拖动，拖动至合适大小后释放鼠标左键，修改按钮名称为“生成工资条”，然后单击工作表中其他任意位置退出该按钮的编辑状态。打开【指定宏】对话框，选择【宏名】列表框中的“生成工资条”选项，单击【确定】按钮，完成宏的指作定操。如图 16-40 所示。



图 16-39　计算实际工资

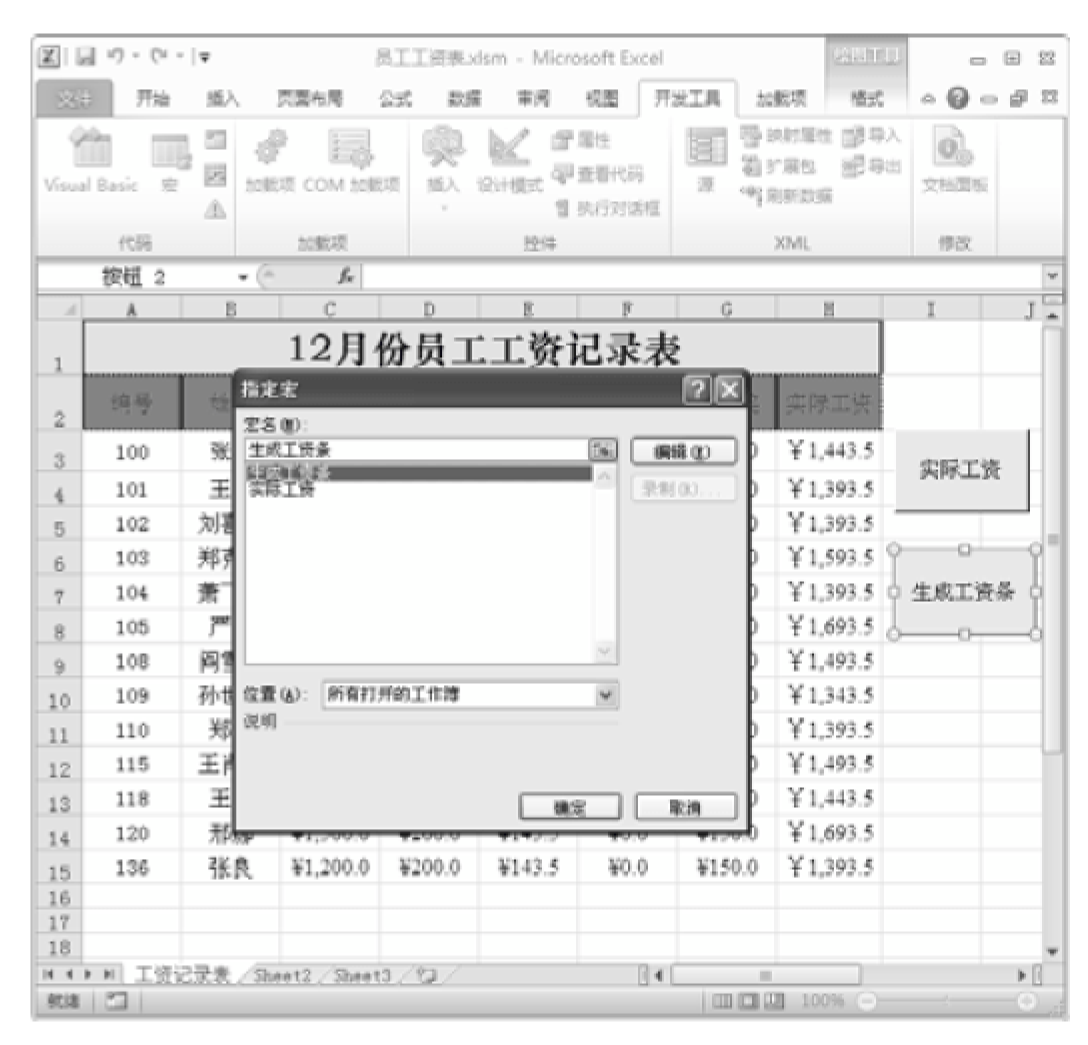


图 16-40　【指定宏】对话框

绑定宏成功之后，单击【生成工资条】按钮，即可运行“生成工资条”的代码，当工资条的生成完成后会弹出如图 16-41 所示的提示信息。单击【确定】按钮时即可看到运行结果，如图 16-42 所示。

提示信息

成功生成员工工资条!

确定

图 16-41　提示工资条已生成

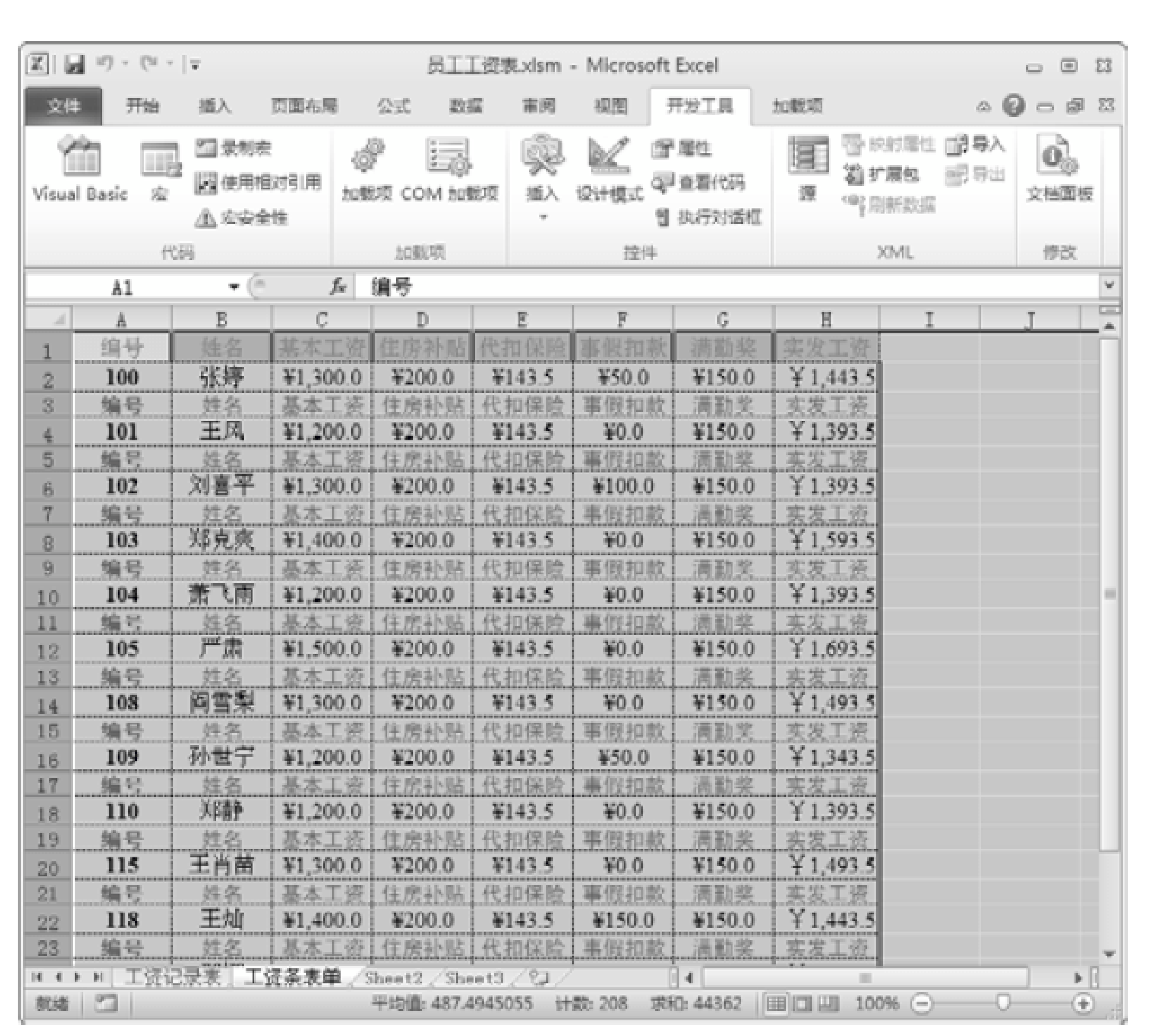


图 16-42　生成的员工工资条

16.2.3　打印员工考核表

员工考核表包含了员工的各项评定内容。使用 VBA 可以实现考核表的整体或部分区域打印。下面将介绍使用 VBA 代码实现区域打印的方法，具体操作步骤如下。



01 打印考核表之前，要生成一个包含考核信息的数据表。可以打开本章提供的一个考核表样例，打开之后的内容如图 16-43 所示，在该工作簿中包含了"员工考核成绩表"工作表，该工作表需要对员工的各项成绩进行综合评定。

02 准备好考核表之后，切换到【开发工具】选项卡，在【控件】选项组中单击【插入】下三角按钮，单击控件列表中的【命令按钮】按钮 （ActiveX 控件），当鼠标指针变成"+"形状时，在工作表中的合适位置，按住鼠标左键不放，拖动至合适大小后释放鼠标左键。在其【属性】对话框中切换到【按字母序】选项卡，将"名称"和 Caption 属性值都修改为"统计总成绩"，如图 16-44 所示。

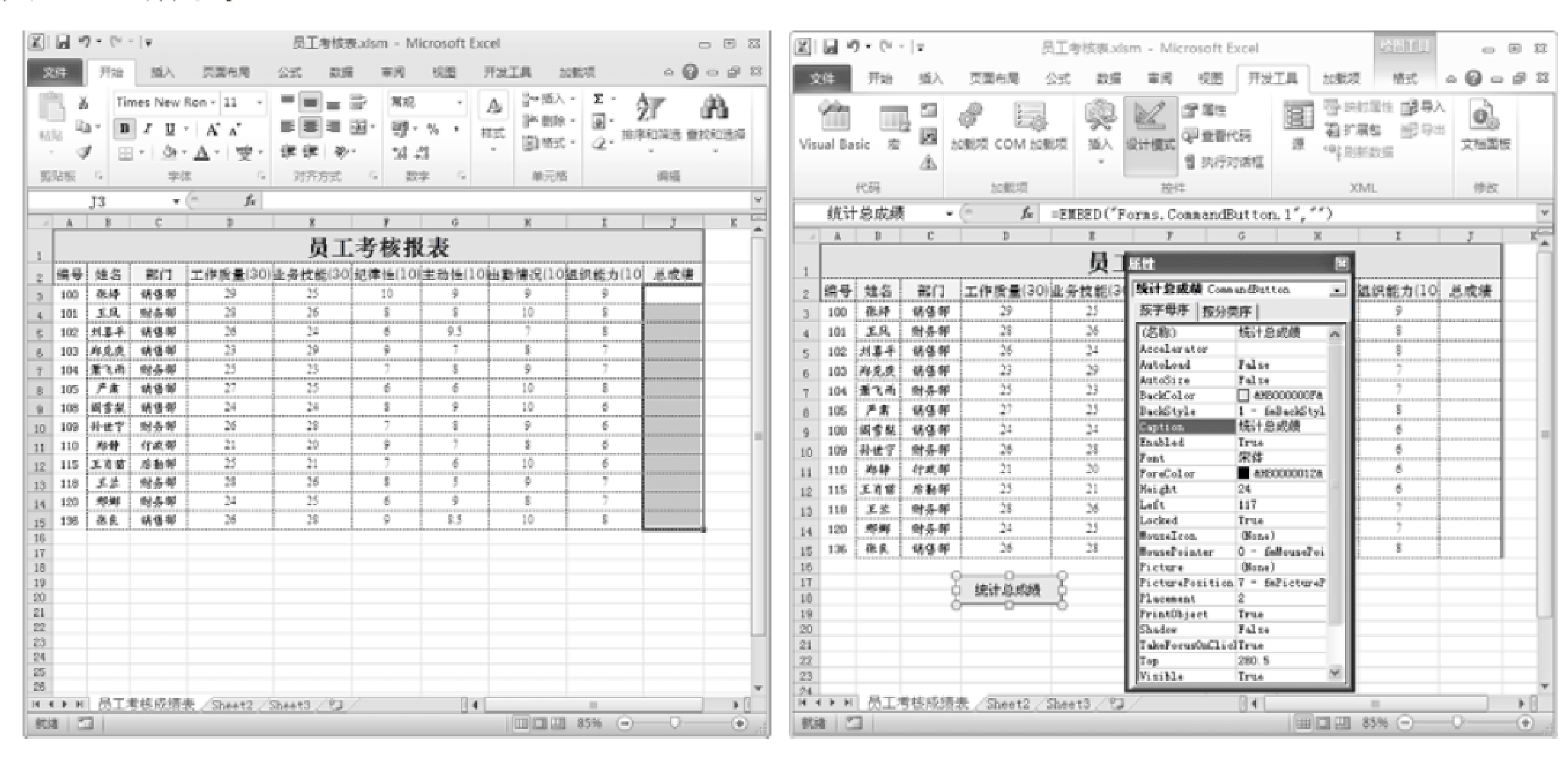


图 16-43　员工考核成绩表　　　　图 16-44　添加命令按钮控件

03 编写【统计总成绩】按钮的 Click 事件代码，在设计模式下，双击命令按钮，进入按钮的代码编辑窗口，输入下面的 VBA 代码。

```
Private Sub 统计总成绩_Click()
    Dim row As Integer
    Dim rowCount As Integer
    rowCount = Sheets("员工考核成绩表").[A1].CurrentRegion. _
    Rows.Count
        For row = 3 To rowCount
            Cells(row, 10) = Cells(row, 4) + Cells(row, 5) + Cells(row, 6) _
            + Cells(row, 7) + Cells(row, 8) + Cells(row, 9)  '计算综合成绩
            With Cells(row, 10)
                .HorizontalAlignment = xlCenter '设置居中显示
                .VerticalAlignment = xlCenter
                .Font.ColorIndex = 9 '将字体颜色设置为“红色”
                .Font.Name = "Times New Roman" '设置字体
                .Font.Size = 11 '设置字号
            End With
        Next row
End Sub
```

代码输入完毕，再次单击工具栏上的【设计模式】按钮，退出按钮的编辑状态，返回“员工考核成绩表”工作表，在该表中单击【统计总成绩】按钮即可完成“总成绩”的计算，如图 16-45 所示。

04 按照同样的方法，在工作表中添加【打印所选区域】按钮，如图 16-46 所示。



图 16-45　统计总成绩　　　　图 16-46　添加【打印所选区域】按钮

添加完【打印所选区域】按钮之后，双击按钮进入该按钮的 Click 事件代码编辑窗口，输入如下 VBA 程序代码。

```
Private Sub 打印所选区域_Click()
    Dim rag As Range
    Dim rowCount As Integer
    Dim colCount As Integer
    Set rag = Application.Selection
    rag.Copy              '进行复制操作
    Worksheets.Add        '插入新工作表
    ActiveSheet.Name = "打印"                    '新工作表命名为“打印”
    Selection.PasteSpecial Paste:=xlPasteAll        '进行选择性粘贴

    rowCount=ActiveSheet.[A1].CurrentRegion.Rows.Count    '统计当前工作表的行数
    colCount=ActiveSheet.[A1].CurrentRegion.Columns.Count'统计当前工作表的列数

    With ActiveSheet
        .Range(.Cells(1, 1), .Cells(rowCount, colCount)).Select
    End With

    Selection.PrintOut   '对选择区域执行打印操作
    Application.DisplayAlerts = False     '取消显示特定的警告和消息
    ActiveWindow.SelectedSheets.Delete    '删除新插入的工作表
End Sub
```

05 代码输入完毕，再次单击工具栏上的【设计模式】按钮，退出按钮的编辑状态，返回工作表，此时用户可以打印所选的区域了。例如想要打印所有“财务部”员工的考核信息，可以在按住 Ctrl 键的同时使用鼠标依次选择相应的行，然后单击【打印所选区域】按钮即可执行打印操作，如图 16-47 所示。



图 16-47　打印选择区域

第 17 章　员工薪金管理系统

员工薪金的管理是指对员工收入与支出的各个方面的管理。员工薪金管理系统一般具有汇总考勤记录、计算员工薪金、员工薪金查询和访问权限控制等功能。本章将通过一个实例来介绍如何使用 Excel VBA 来设计一个基本的员工薪金管理系统。



17.1 设置员工薪金管理工作簿

通过设置员工薪金管理工作簿，用户可以访问工作簿中各个工作表、汇总员工的考勤记录、计算员工的薪金等。

17.1.1 添加美观的按钮

在工作簿中添加一些自定义的按钮，可以方便查询和管理工作表。这些自定义的按钮不仅可以实现快速访问工作表，而且还可以使管理界面更加美观。

1. 在员工薪金管理系统的首页工作表中添加按钮

管理系统包含了多个作用不同的工作表，如果非常多的话，在 Excel 工作簿中单击下面的工作表标签进行切换会不方便，而使用 VBA，用户可以添加一些更加直接、快捷的访问按钮来访问这些工作表。下面来看如何创建这些访问按钮。

01 打开本章的样例工作簿〝员工薪金管理 1〞，并选择第一个〝首页〞工作表，结果如图 17-1 所示。

首先来添加第一个图形按钮。在“首页”工作表中切换到【插入】选项卡，在【插图】选项组的【形状】下拉列表中单击【矩形】组中的圆角矩形按钮▢，然后在“首页”工作表中绘制圆角矩形，如图 17-2 所示。



图 17-1 “员工薪金管理 1”工作簿

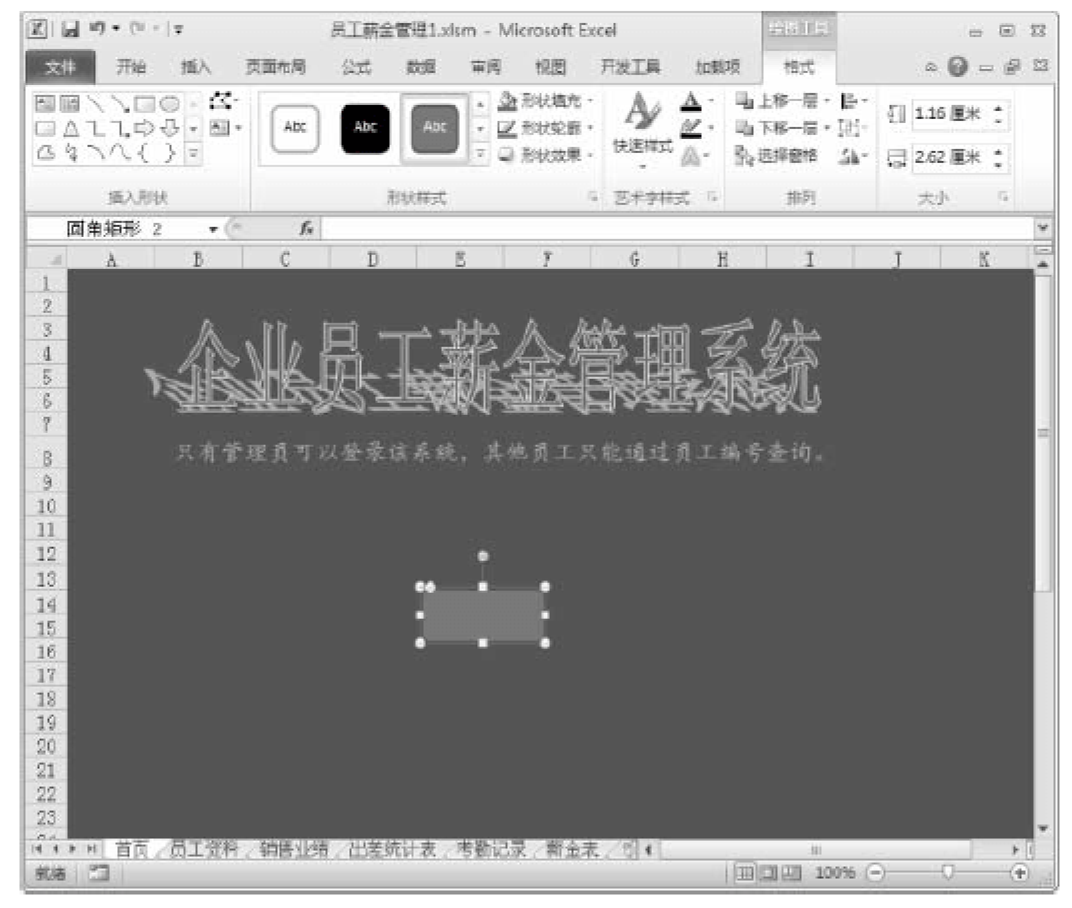


图 17-2 绘制圆角矩形

右击添加的圆角矩形，选择弹出菜单中的【设置形状格式】命令，打开【设置形状格式】对话框，在【填充】选项卡中设置该圆角矩形的填充样式，如图 17-3 所示。设置完成后，单击【关闭】按钮保存对该圆角矩形样式的设置。

再次右击添加的圆角矩形，选择弹出菜单中的【编辑文字】命令，在圆角矩形上的文本框中输入“员工资料”，同时设置字体为“华文中宋”，字号为 14，颜色为“深蓝”，并单击【加粗】按钮 **B**，其效果如图 17-4 所示。



图 17-3 设置圆角矩形的填充样式

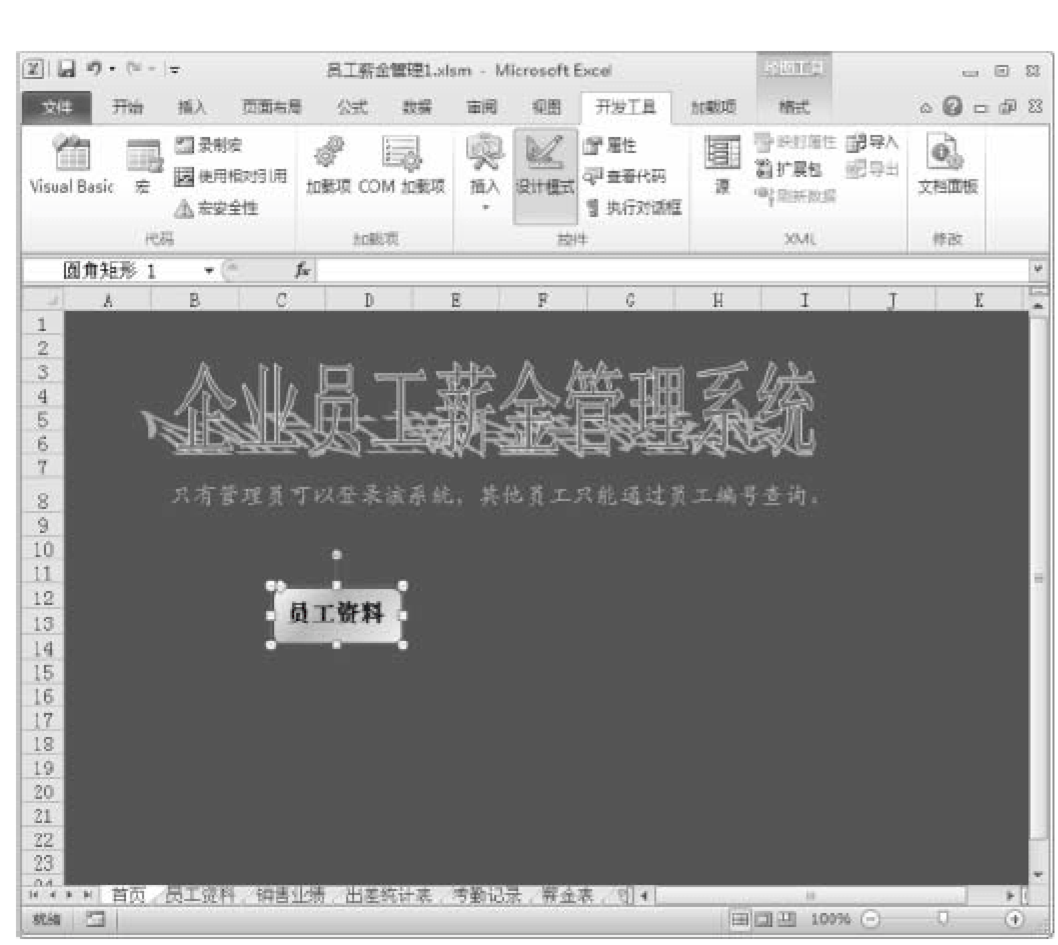


图 17-4 添加文本

02 按照步骤 1 的操作过程，在“首页”工作表中添加另外的几个圆角矩形，如图 17-5 所示。

03 添加完图形按钮，下面来为这些按钮指定宏操作。首先，右击名称为“员工资料”的按钮，在弹出的快捷菜单中选择【指定宏】菜单命令，打开【指定宏】对话框，在【宏名】文本框中输入“显示员工资料”，如图 17-6 所示。

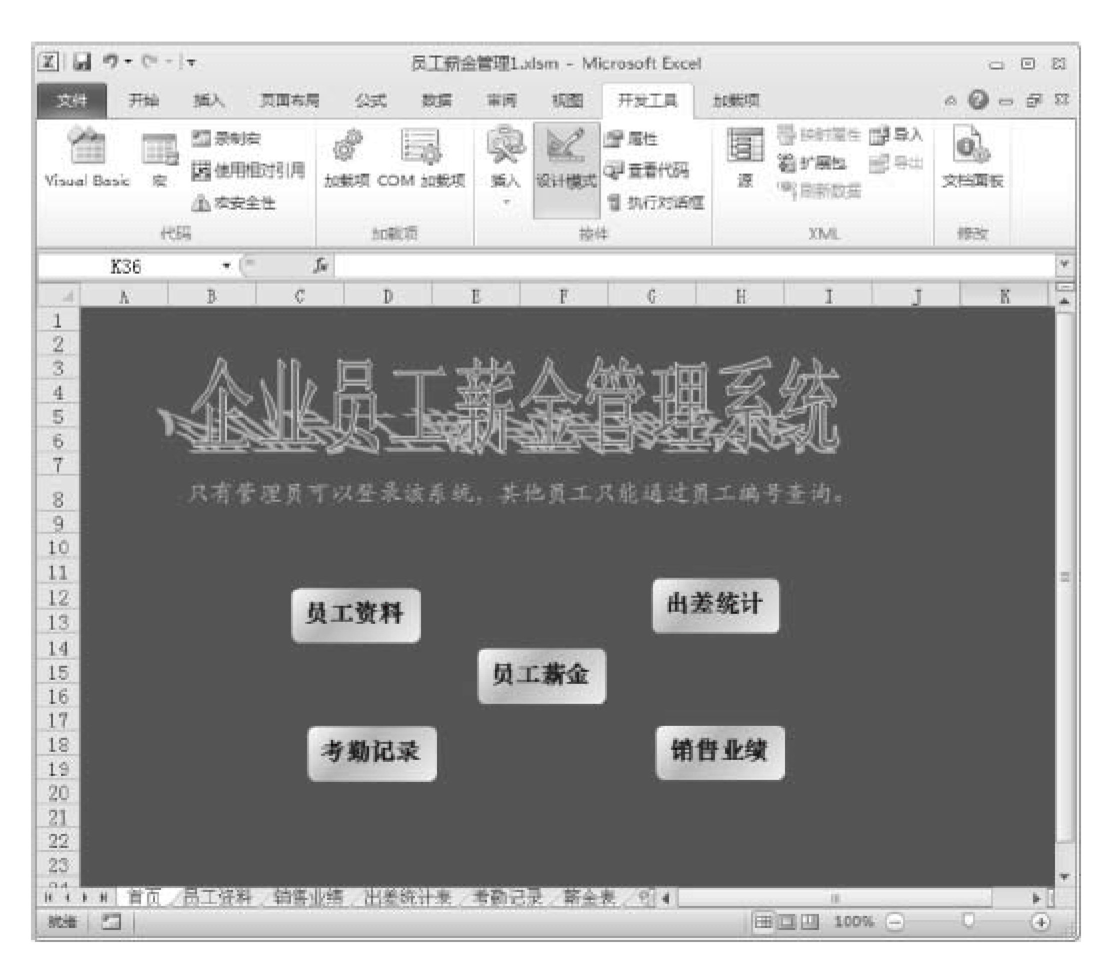


图 17-5　添加其他几个圆角矩形

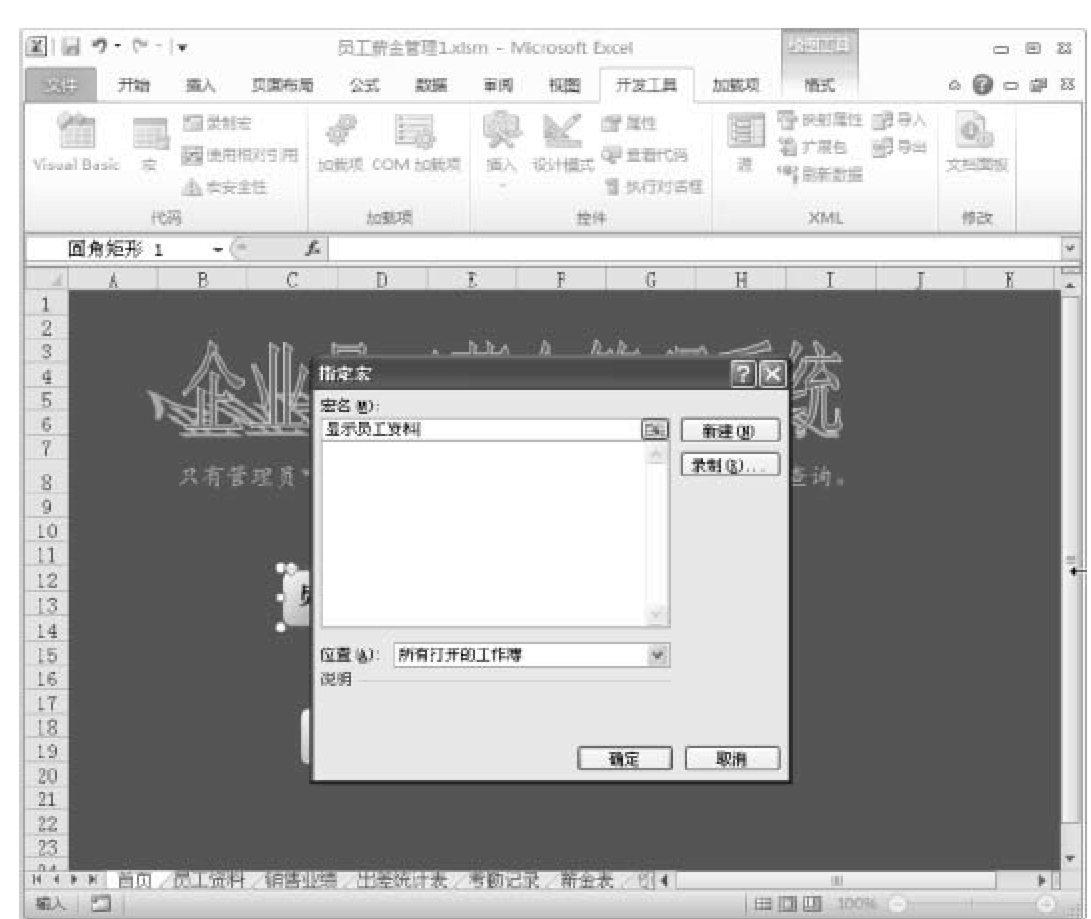


图 17-6　【指定宏】对话框

单击【新建】按钮，即可打开 VBA 代码编辑窗口，并且系统会自动新建一个名称为“模块 1”的模块，在模块 1 的代码窗口中显示指定宏“显示员工资料”的代码框架，如图 17-7 所示。

在该代码框架中输入完整的代码程序，代码如下：

```
Sub 显示员工资料()
Sheets("员工资料").Activate
End Sub
```

04 关闭 VBA 代码窗口返回到“首页”工作表中，此时名称为“员工资料”的圆角矩形已成为一个按钮，单击该按钮可以调用“显示员工资料”子过程。切换到“员工资料”工作表中，把鼠标放在该按钮上时，可以看到鼠标形状变为👆，如图 17-8 所示。

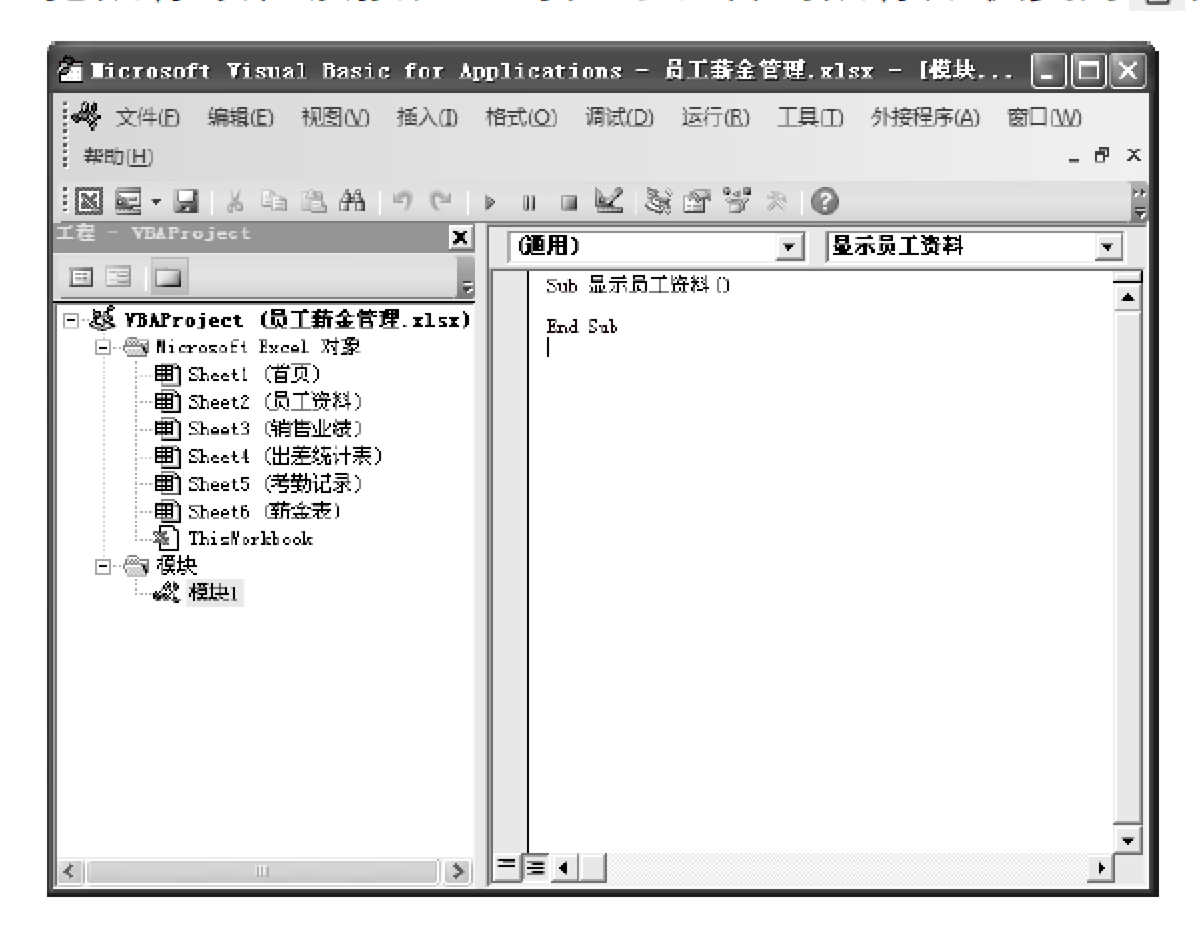




图 17-7　“显示员工资料”的代码框架



图 17-8　【员工资料】按钮

05 按 Alt+F11 组合键，打开 VBA 代码窗口，在模块 1 的代码编辑窗口中依次添加下面几个子过程代码：

```
Sub 显示员工资料()
    Sheets("员工资料").Activate
End Sub

Sub 显示销售业绩()
    Sheets("销售业绩").Activate
End Sub

Sub 显示出差统计()
    Sheets("出差统计表").Activate
End Sub

Sub 显示考勤记录()
    Sheets("考勤记录").Activate
End Sub

Sub 显示员工薪金()
    Sheets("薪金表").Activate
End Sub

Sub 返回()
    Sheets("首页").Activate
End Sub
```

06 输入完毕后，返回"首页"工作表，依次为每个图形按钮指定操作宏。右击名为"出差统计"的圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，在【宏名】列表框中选择"显示出差统计"选项，如图 17-9 所示。

单击【确定】按钮完成指定宏的操作。按照同样的方法为其他圆角矩形指定模块 1 中已经编辑好的宏。

2. 在其他工作表中添加【返回】按钮

完成了上面的各个按钮的创建之后，现在可以方便、快捷地访问各个工作表了，但是现在还有一个问题，既然可以直接访问这些表单，那能不能从这些工作表中直接返回到"首页"工作表呢？当然可以。与前面的方法相同，只要在需要返回的工作表中添加一个按钮和一小段代码就可以实现了。下面来看【返回】按钮的创建过程。

01 打开"员工薪金管理 1"工作簿之后，按 Alt+F11 组合键，打开 VBA 代码窗口，在模块 1 的代码编辑窗口中添加下面的 VBA 代码：

```
Sub 返回()
Sheets("首页").Activate
End Sub
```

该段代码用来返回到"首页"工作表。输入完毕后关闭 VBA 代码窗口返回工作表。

02 首先选择在"员工资料"工作表创建按钮。切换到该工作表中，在【插入】选项卡中 【插

图】选项组的【形状】下拉列表中单击【矩形】组中的【圆角矩形】按钮▢，在“员工资料”工作表中绘制一个圆角矩形，如图 17-10 所示。

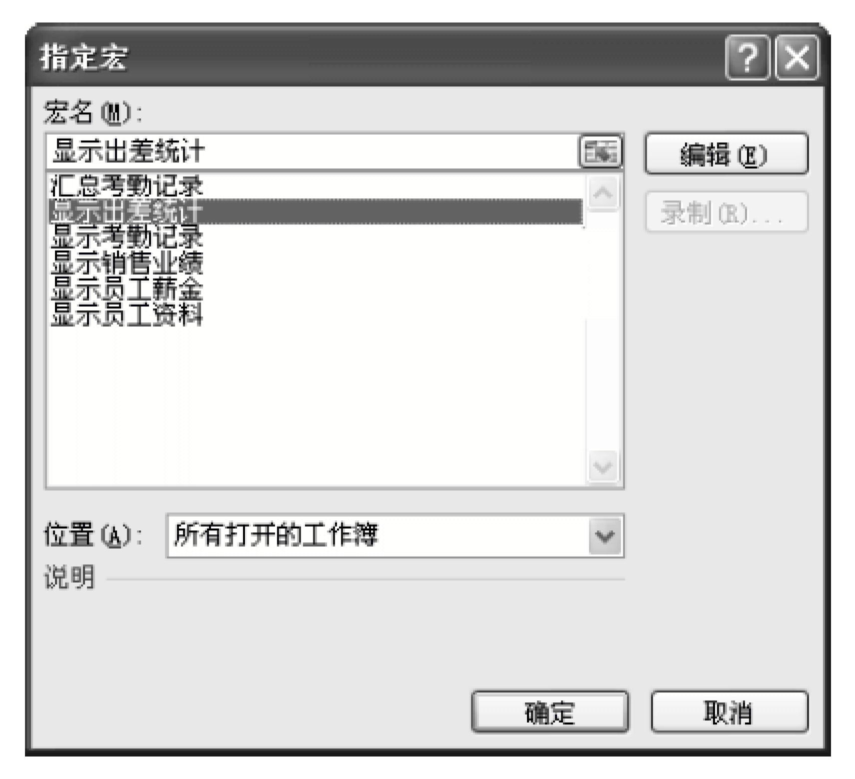


图 17-9　【指定宏】对话框



图 17-10　绘制一个圆角矩形

设置矩形图形的阴影效果。切换到【格式】选项卡，在【形状样式】选项组的【形状效果】下拉列表中选择需要的阴影效果，如图 17-11 所示。

设置阴影效果之后，右击圆角矩形，在弹出的快捷菜单中选择【设置形状格式】命令，打开【设置形状格式】对话框，设置相应的样式，各个参数如图 17-12 所示。

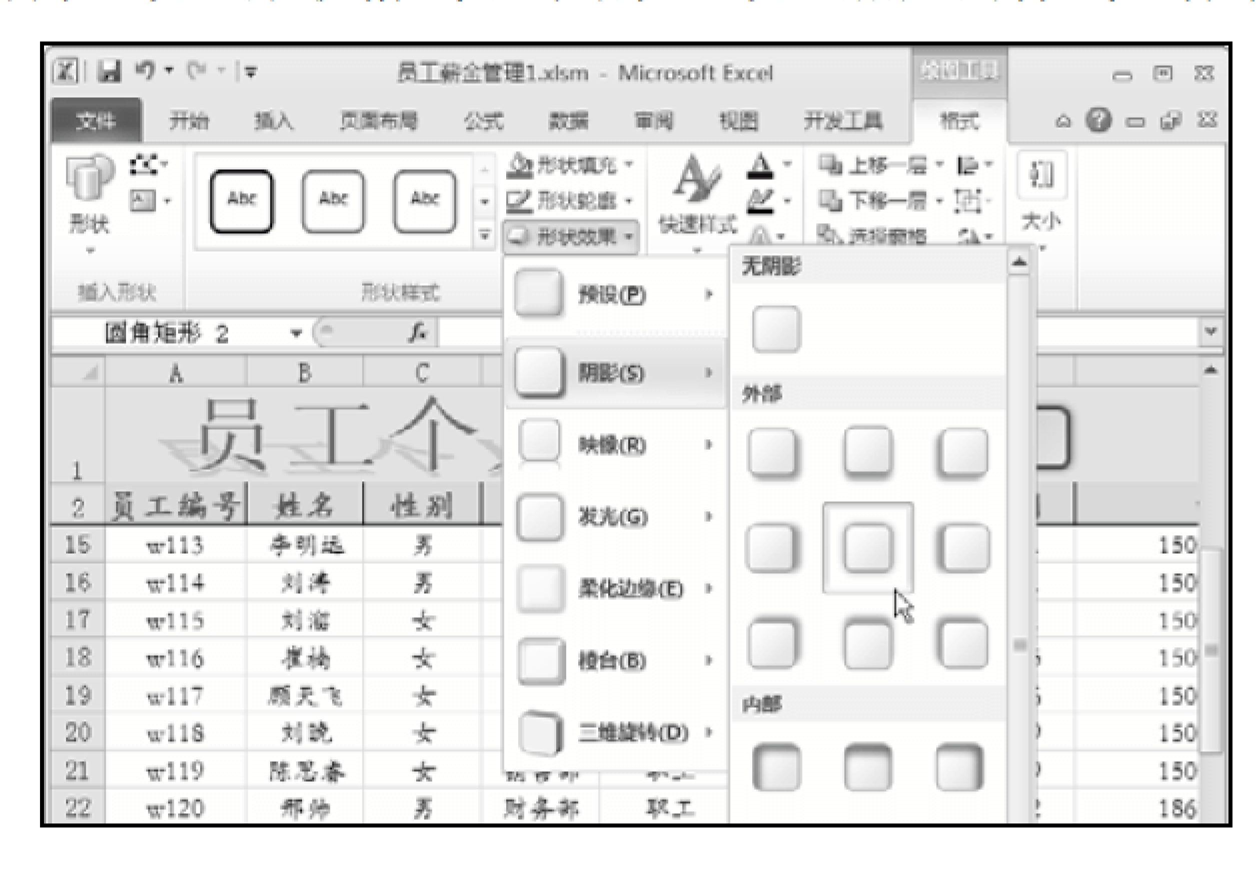


图 17-11　设置阴影效果



图 17-12　【设置形状格式】对话框

单击【设置形状格式】对话框中的【关闭】按钮，完成对圆角矩形的设置。然后在圆角矩形中添加文本“返回”，并设置其字体、字号等，如图 17-13 所示。

03 按照同样的方法在“销售业绩”工作表、“出差统计表”工作表、“考勤记录”工作表及“薪金表”工作表中分别添加包含“返回”文本的圆角矩形，或直接将设置好的圆角矩形复制到其他工作表中。

04 代码编写完了，所有表单中的操作按钮也有了，下面将把这两步中的内容联系到一起完成返回首页的操作。首先切换到“员工资料”工作表，右击【返回】圆角矩形，在弹出的快捷菜单中选择【指定宏】命令，打开【指定宏】对话框，在【宏名】列表框中选择“返回”选项，如图 17-14 所示。



图 17-13　添加文本“返回”



图 17-14　【指定宏】对话框

05 单击【确定】按钮关闭【指定宏】对话框，此时【返回】圆角矩形已经成为一个按钮，单击该【返回】按钮，即可返回“首页”工作表，如图 17-15 所示。

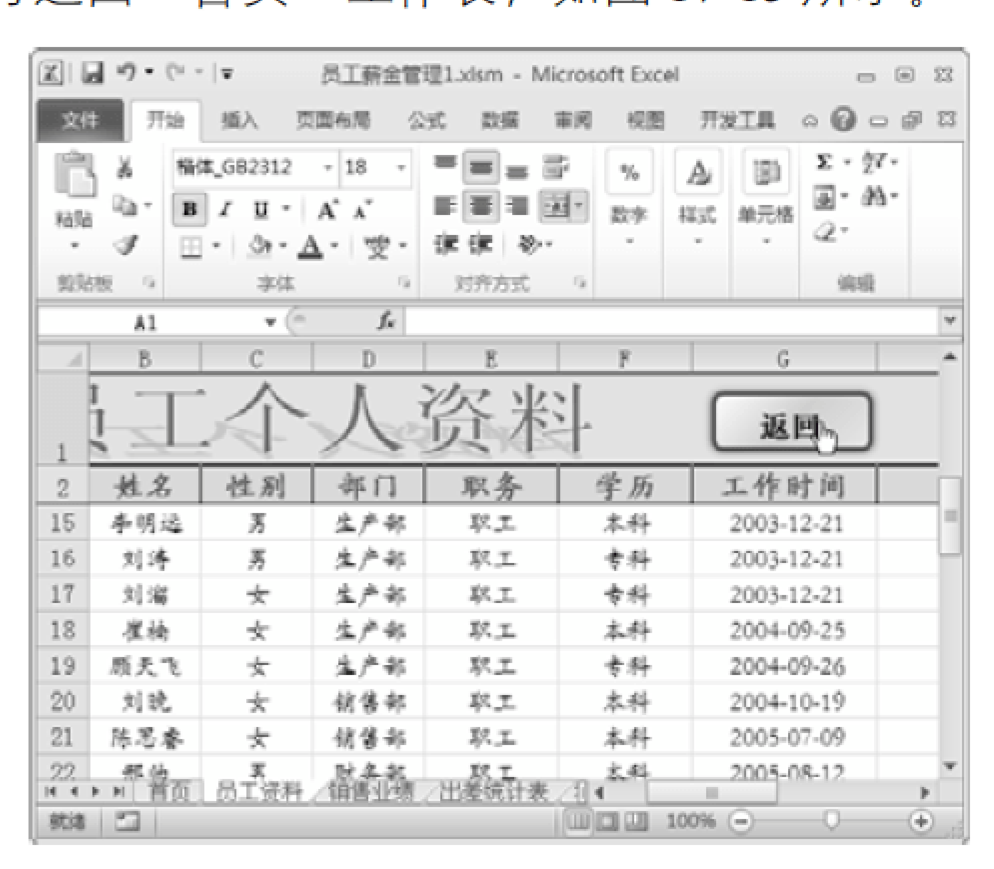


图 17-15　返回“首页”工作表

06 用同样的方法为其他工作表中的按钮指定返回首页操作的宏。

17.1.2 汇总考勤记录

考勤信息包括员工的迟到/早退次数、旷工次数和加班次数等，并且每个员工的出勤情况不同，所以“考勤记录”工作表中的数据是变化的。根据实际情况，汇总考勤记录的具体操作步骤如下。

01 打开本章的“员工薪金管理 2”工作簿，切换到“考勤记录”工作表中，在【插入】选项卡的【插图】选项组的【形状】下拉列表中单击【矩形】组中的【圆角矩形】按钮▢，在工作表中绘制一个圆角矩形，按照前面介绍的过程，添加【汇总考勤记录】按钮，并为按钮添加样式、文字和字号，修改后的效果如图 17-16 所示。

02 按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中创建汇总考勤记录子过程，输入代码如下。

```
   Sub 汇总考勤记录()
       Dim row As Integer, col As Integer
       Dim chidao As Integer, zaotui As Integer, kuanggong As Integer, jiaban As
Integer
       Dim count As Integer

       count = Sheets("考勤记录").Range("A1").CurrentRegion.Rows.count
       For row = 4 To count
          chidao = 0
          zaotui = 0
          kuanggong = 0
          jiaban = 0

          For col = 2 To 32
              If InStr(Cells(row, col), "迟到") > 0 Then
                  chidao = chidao + 1
              End If
              If InStr(Cells(row, col), "早退") > 0 Then
                 zaotui = zaotui + 1
              End If
              If InStr(Cells(row, col), "旷工") > 0 Then
                 kuanggong = kuanggong + 1
              End If
              If InStr(Cells(row, col), "加班") > 0 Then
                 jiaban = jiaban + 1
              End If
          Next col

              Cells(row, 33) = chidao
          Cells(row, 34) = zaotui
          Cells(row, 35) = kuanggong
          Cells(row, 36) = jiaban
       Next row
   End Sub
```

该段代码实现获取每位员工在 12 月份每一天的考勤记录。如果某位员工在某一天迟到，则将这位员工相应的迟到总次数加 1。相应地，早退、旷工和加班次数也是这样统计的。

03 关闭 VBA 代码编辑窗口，返回“考勤记录”工作表中，右击【汇总考勤记录】圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，在【宏名】列表框中选择“汇总考勤记录”，如图 17-17 所示。单击【确定】按钮，完成指定宏操作。

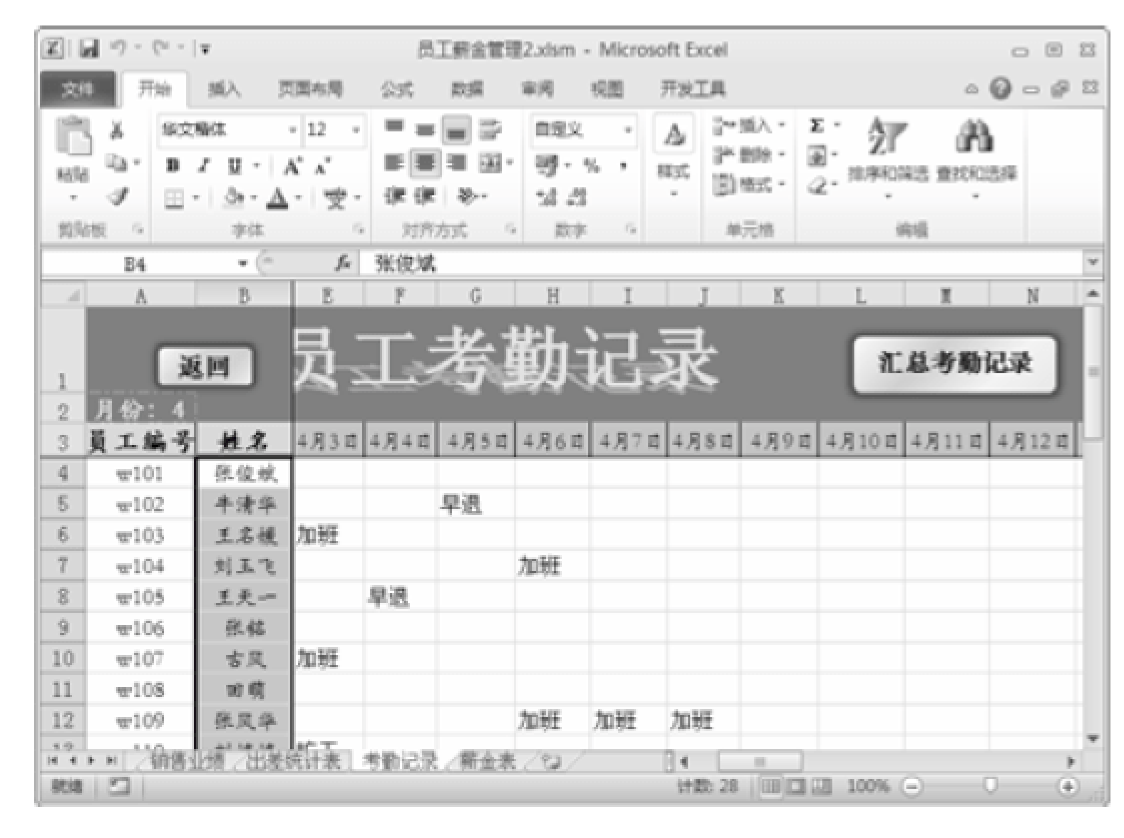



图 17-16　输入文本并设置其效果



图 17-17　为按钮指定宏

此时【汇总考勤记录】圆角矩形已经成为一个按钮，单击【汇总考勤记录】按钮，然后将工作表拉到最后列，即可看到计算出的每位员工的迟到次数、早退次数、旷工次数和加班次数，如图 17-18 所示。

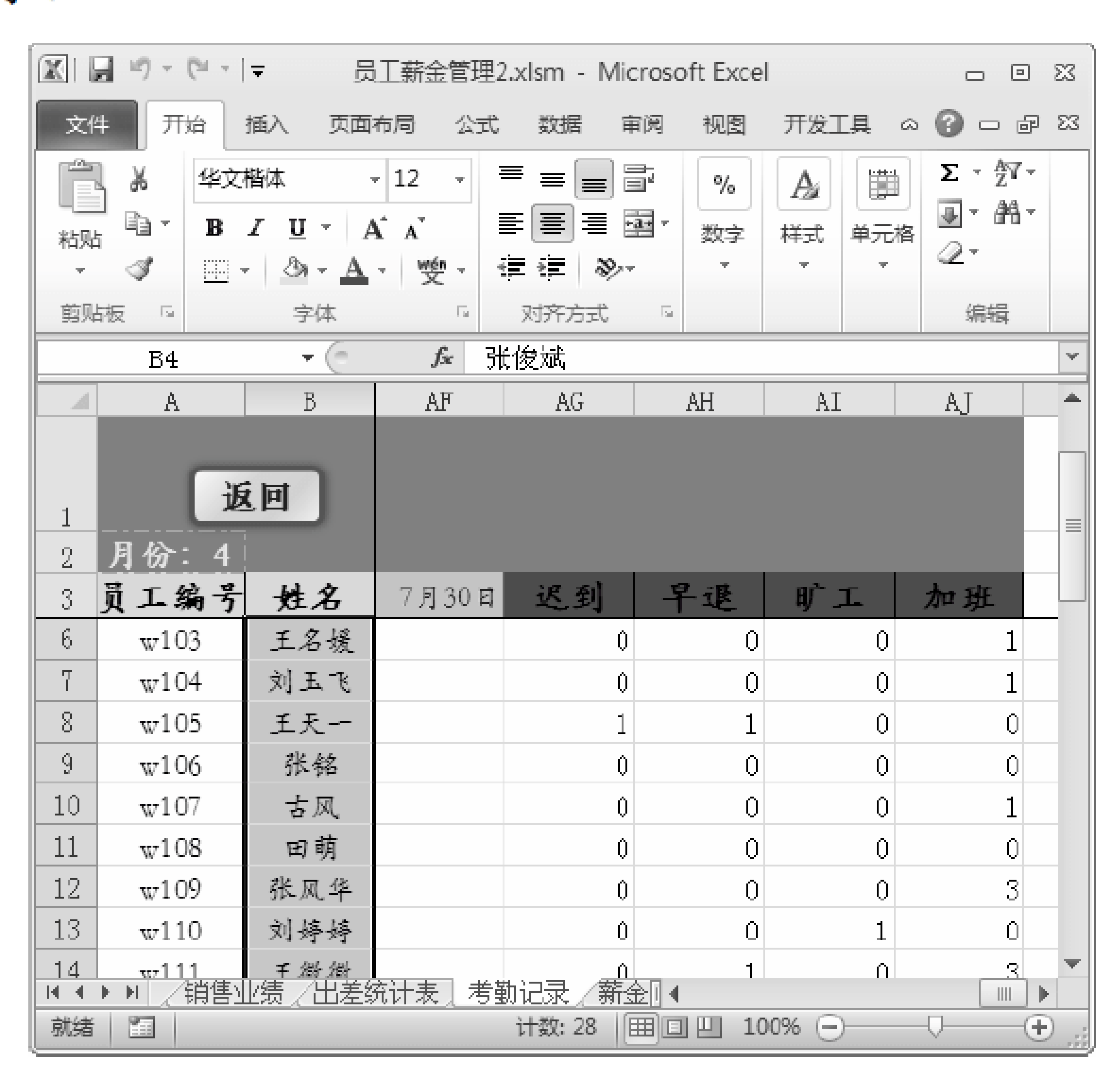


图 17-18　计算考勤结果

17.1.3　计算员工薪金

员工薪金包括基本工资、补助、奖金、加班费等。本小节将介绍如何使用 VBA 计算员工的实发工资，具体操作步骤如下。

01 打开本章的"员工薪金管理 3"工作簿，切换到"薪金表"工作表中，如图 17-19 所示。按照前面"考勤记录"工作表中添加"汇总考勤记录"圆角矩形的方法，在"薪金表"工作表中添加图 17-20 所示的 8 个圆角矩形，并设置其文本字体为"华文中宋"、字号为 10 号。

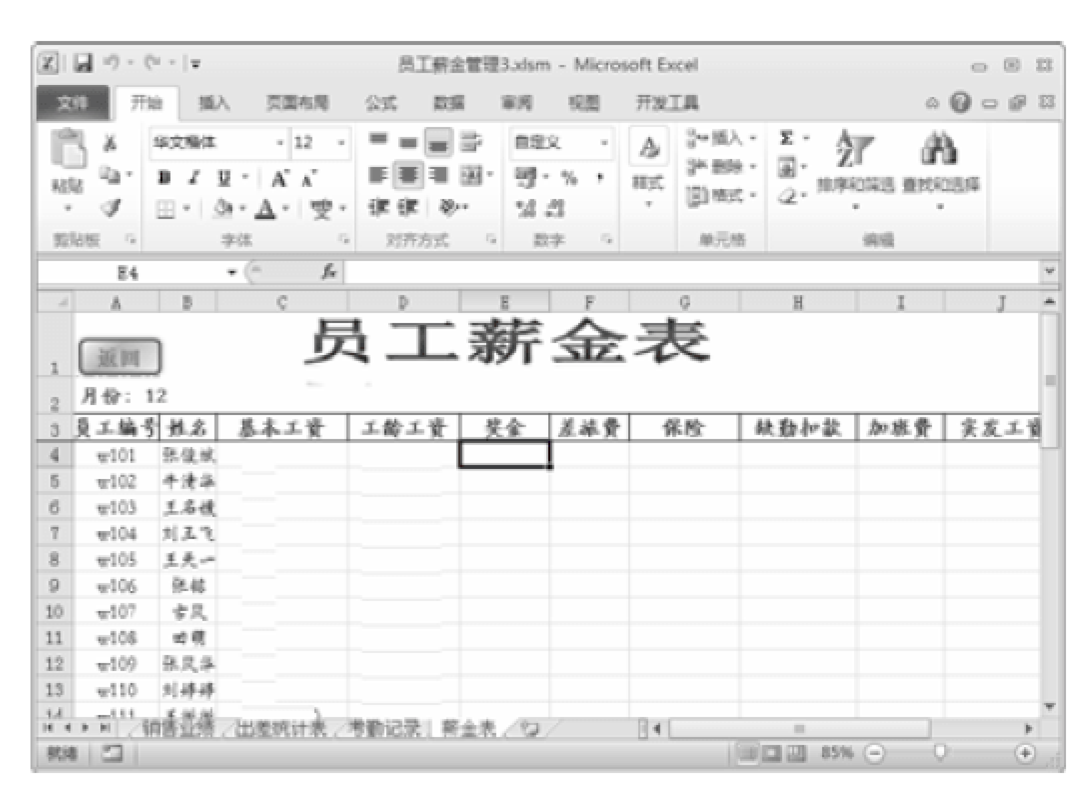


图 17-19　“薪金表”工作表

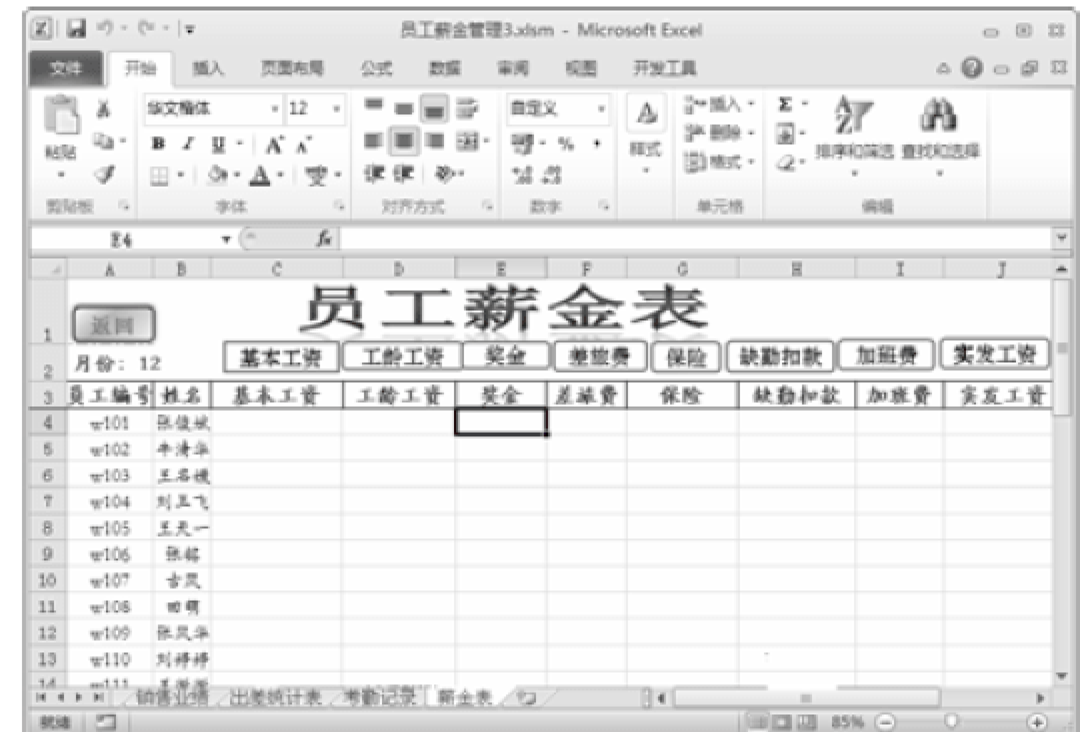


图 17-20　添加 8 个圆角矩形

基本工资与员工的学历有关，学历越高其基本工资也越高。这里假设员工的基本工资设定标准如表 17-1 所示。

表 17-1　员工的基本工资设定标准

学历	博士	硕士	本科	专科
基本工资	2800	2500	1800	1500

02 按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中编写“基本工资”子过程的代码。

```
Sub 基本工资()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    count1 = Sheets("员工资料").Range("A1").CurrentRegion.Rows.count
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count2
       For col = 3 To count1
          If Sheets("员工资料").Cells(col, 1) = Cells(row, 1) Then
     '在“员工资料”工作表中找到相应的员工
            Select Case Sheets("员工资料").Cells(col, 4)'根据员工的学历制定基本工资
                Case "专科"
                    Cells(row, 3) = 1500
                Case "本科"
                    Cells(row, 3) = 1800
                Case "硕士"
                    Cells(row, 3) = 2500
                Case "博士"
                    Cells(row, 3) = 2800
             End Select
             Exit For
         End If
       Next col
    Next row
End Sub
```

这段代码首先使用变量 count1 和 count2 分别记录“员工资料”工作表和“薪金表”工作表的当前可见单元格的行数，然后使用嵌套 For 循环遍历这两个工作表中的所有员工记录行。在 For 循环中使用 If 语句判断这两个工作表中当前员工记录行的员工编号是否相同，如果相同则根据“员工资料”工作表中员工的学历计算该员工在“薪金表”工作表中的基本工资。

03 关闭 VBA 代码编辑窗口，返回“薪金表”工作表中，右击【基本工资】圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，在【宏名】列表框中选择“基本工资”选项，如图 17-21 所示。

单击【确定】按钮，完成【基本工资】按钮的指定宏操作。此时【基本工资】圆角矩形将作为一个按钮，单击该按钮，可以计算出每位员工的基本工资，如图 17-22 所示。

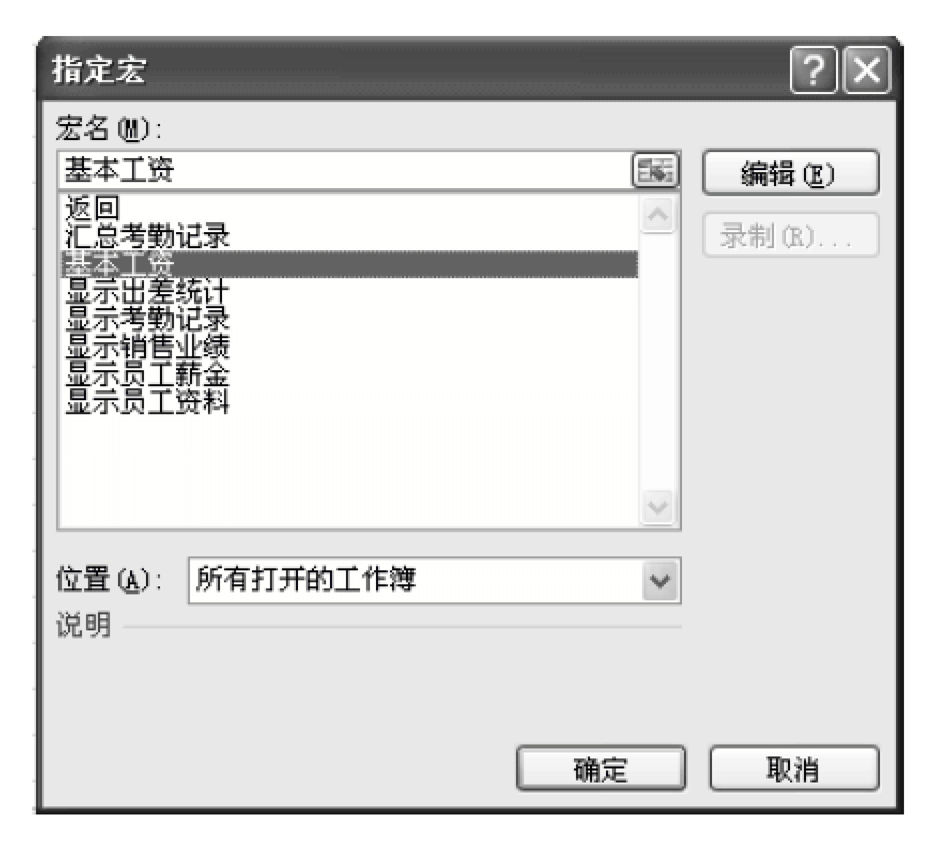


图 17-21　指定宏名为“基本工资”

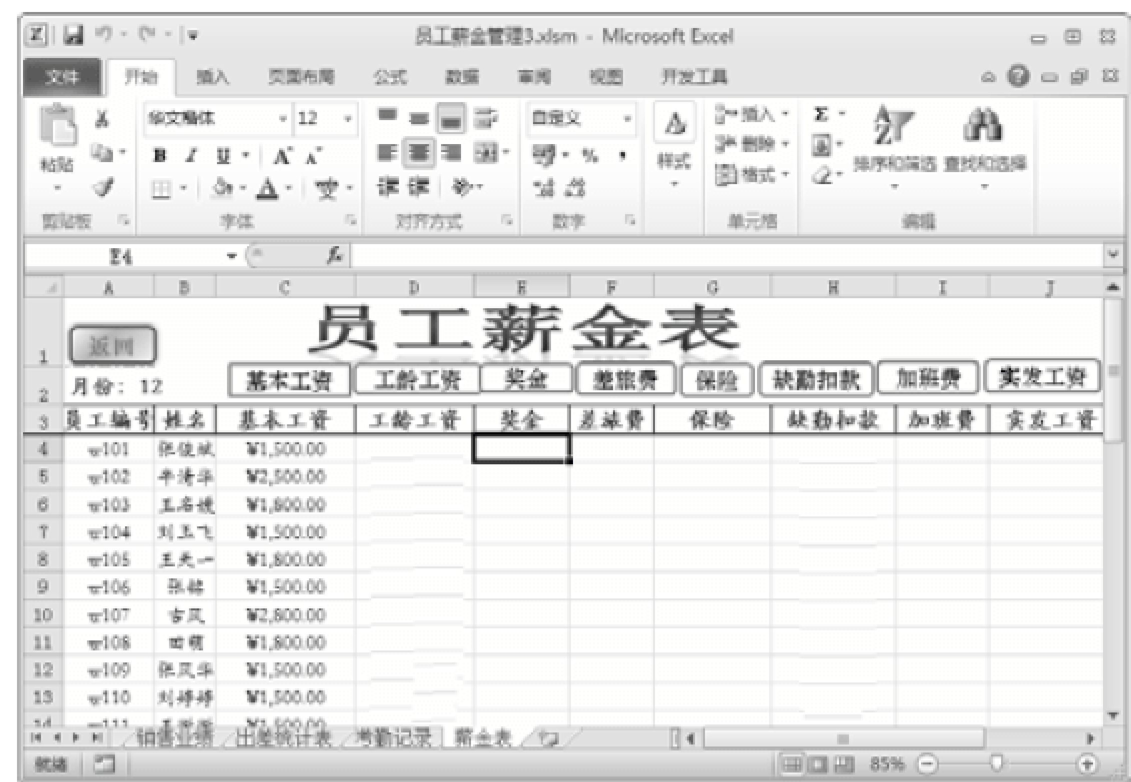


图 17-22　计算出每位员工的基本工资

04 接下来计算员工的工龄工资，员工工作年限越长，工龄工资越高。假设员工工龄每增加 1 年，工龄工资就增加 200 元。如果工龄超过 10 年，则工龄工资统一设定为 2000 元。按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中输入计算工龄工资的程序代码。

```
Sub 工龄工资()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    Dim year_num As Integer  '存储工龄
    count1 = Sheets("员工资料").Range("A1").CurrentRegion.Rows.count
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count2
       For col = 3 To count1
          If Sheets("员工资料").Cells(col, 1) = Cells(row, 1) Then
            '在“员工资料”工作表中找到相应的员工
            year_num = Year(Now) - Year(Sheets("员工资料").Cells(col, 7))
            '获取当前日期和员工工作日期的年份之差
            If year_num <= 10 Then  '如果工龄不大于 10
                Cells(row, 4) = 200 * year_num
            Else
               Cells(row, 4) = 2000
            End If
```

```
            Exit For
         End If
       Next col
    Next row
End Sub
```

输入完毕，关闭 VBA 代码窗口，返回“薪金表”工作表中，右击【工龄工资】圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，选择【宏名】列表框中的“工龄工资”选项，如图 17-23 所示。

单击【确定】按钮，完成【工龄工资】按钮的指定宏操作。此时【工龄工资】圆角矩形将作为一个按钮，单击该按钮，即可计算出每位员工的工龄工资，如图 17-24 所示。

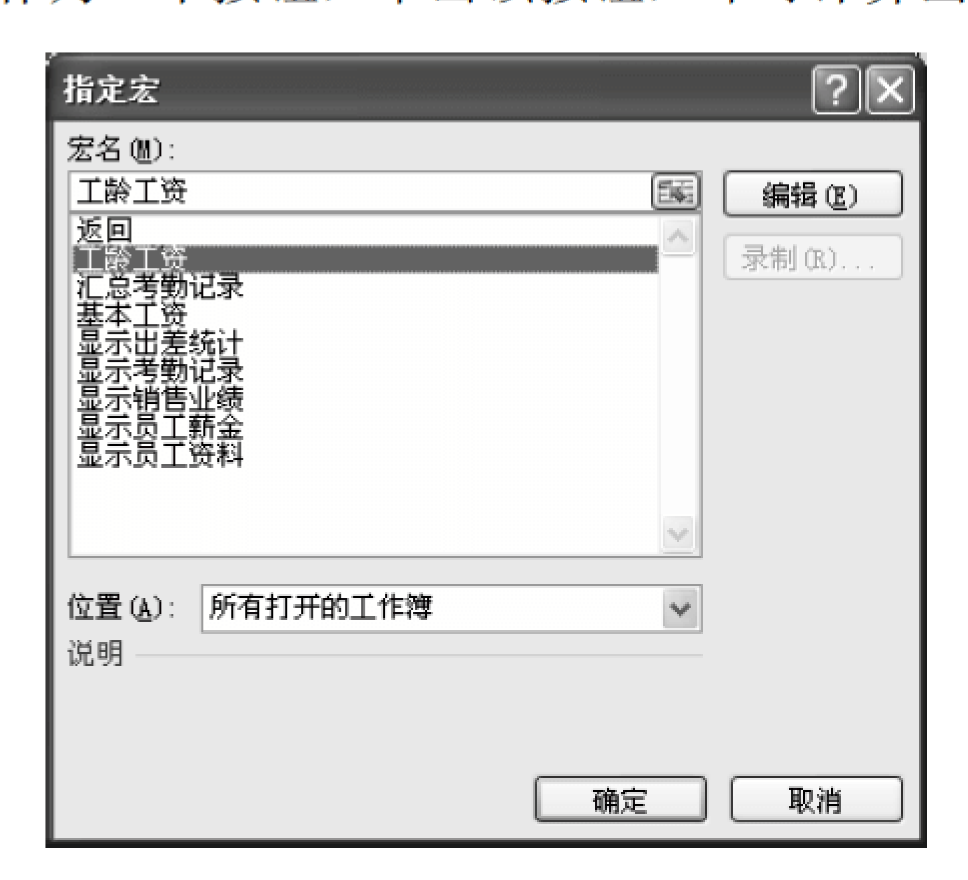


图 17-23　指定宏名为“工龄工资”

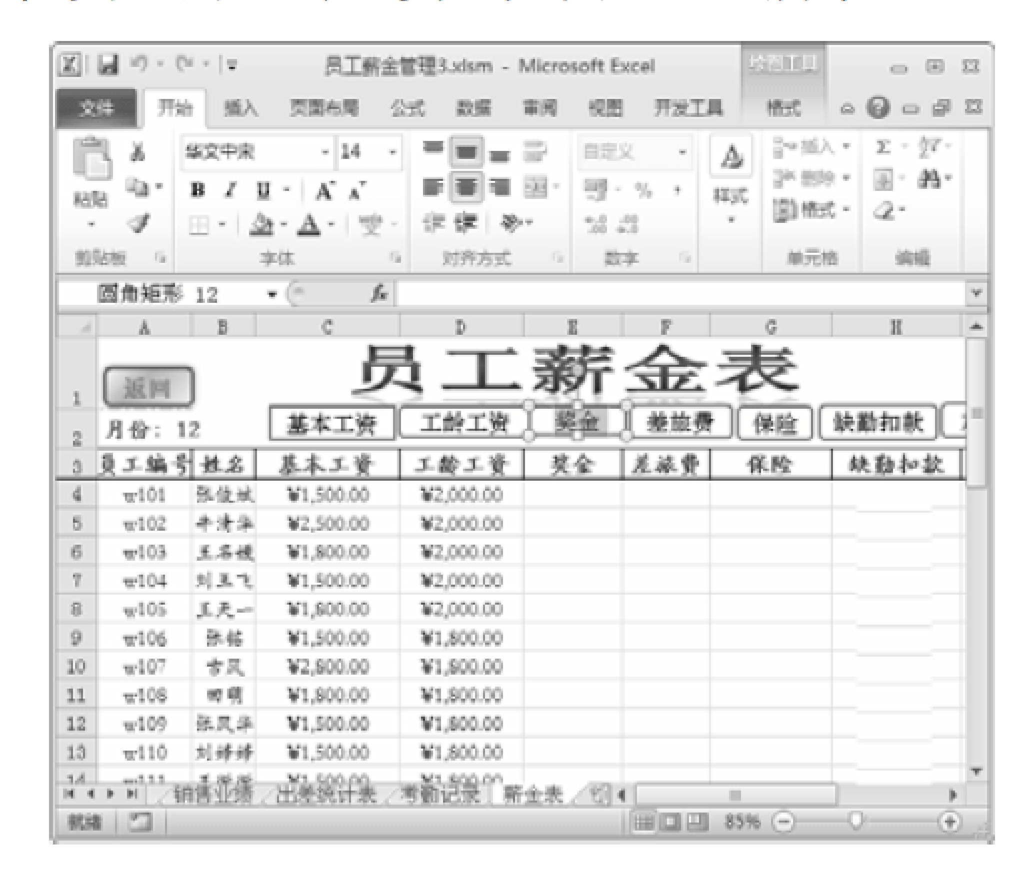


图 17-24　计算出每位员工的工龄工资

05 计算员工的奖金，这里假定员工的奖金主要指销售人员的奖金，奖金额度为销售额的 5%。该实例以计算销售人员的奖金的方法为例，将其他部门员工的奖金设置为 0。按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中创建计算奖金的子过程，输入代码如下。

```
Sub 奖金()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    Dim prize As Double  '奖金
    count1 = Sheets("销售业绩").Range("C3").CurrentRegion.Rows.count + 2
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count1
       For col = 4 To count2 '在“薪金表”工作表中找到销售部员工
          If Cells(col, 1) = Sheets("销售业绩").Cells(row, 3) Then
           prize = Sheets("销售业绩").Cells(row, 5) * 0.05  '计算奖金
           Cells(col, 5) = prize
           Exit For
         End If
       Next col
    Next row
    For col = 4 To count2 '将其他员工的奖金设置为 0
       If Cells(col, 5) = "" Then
```

```
            Cells(col, 5) = 0
        End If
    Next col
End Sub
```

该段代码使用嵌套 For 循环遍历“销售业绩”工作表和“薪金表”工作表，在 For 循环中使用 If 语句判断这两个表中当前员工记录行中的员工编号是否相同，相同表示在“薪金表”工作表中找到了一位销售人员，然后根据“销售业绩”工作表中的该销售人员的销售额来计算奖金。对于其他员工，奖金列单元格的值都为 0。

输入完毕后，关闭 VBA 代码窗口，返回“薪金表”工作表中，右击【奖金】圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，选择【宏名】列表框中的“奖金”选项，如图 17-25 所示。单击【确定】按钮，完成指定宏操作。

此时【奖金】圆角矩形可以作为一个按钮，单击该按钮，即可计算出每位员工的奖金，如图 17-26 所示。



图 17-25　指定宏名为“奖金”

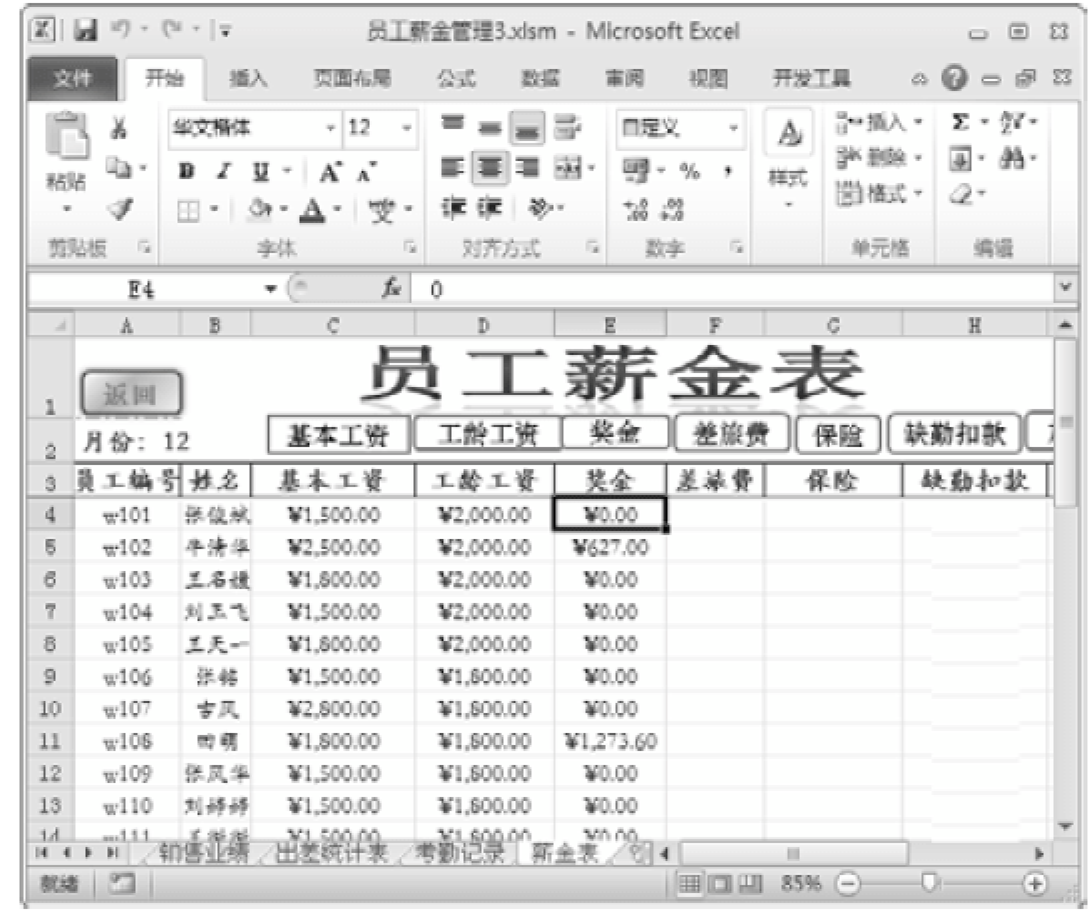


图 17-26　计算出每位员工的奖金

06 计算员工差旅费，许多公司在员工出差时会提供差旅费用补助，这里假设差旅费的计算公式为“差旅费=1000+15*出差天数”，接下来根据上述公式，结合“出差统计表”工作表计算出差员工的差旅费。

按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中创建计算差旅费的子过程，输入代码如下。

```
Sub 差旅费()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    Dim subsidy As Double  '差旅费
    count1 = Sheets("出差统计表").Range("B3").CurrentRegion.Rows.count + 2
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count1
       For col = 4 To count2 '在“薪金表”工作表中找到出差的员工
         If Cells(col, 1) = Sheets("出差统计表").Cells(row, 2) Then  '员工编号相
```

```
                subsidy = 2000 + Sheets("出差统计表").Cells(row, 7) * 15  '计算差旅
费
                Cells(col, 6) = subsidy
                Exit For
            End If
        Next col
    Next row
    For col = 4 To count2 '将其他员工的差旅费设置为 0
        If Cells(col, 6) = "" Then
            Cells(col, 6) = 0
        End If
    Next col
End Sub
```

输入完毕，关闭 VBA 代码窗口，返回“薪金表”工作表中，右击【差旅费】圆角矩形，选择弹出菜单中的【指定宏】命令，打开【指定宏】对话框，选择【宏名】列表框中的“差旅费”选项，如图 17-27 所示。

单击【确定】按钮，完成【差旅费】按钮的指定宏操作。此时【差旅费】圆角矩形可以作为一个按钮，单击该按钮，计算出每位员工的差旅费，如图 17-28 所示。

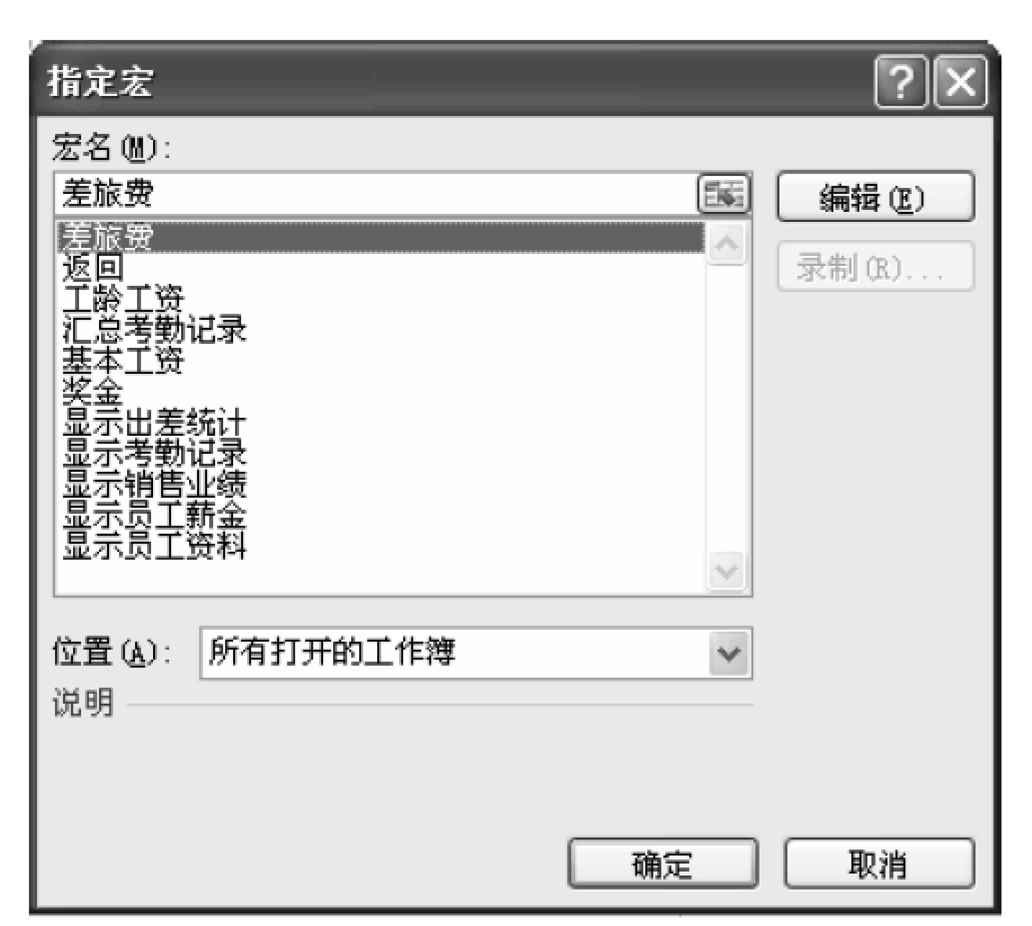


图 17-27 指定宏名为“差旅费”

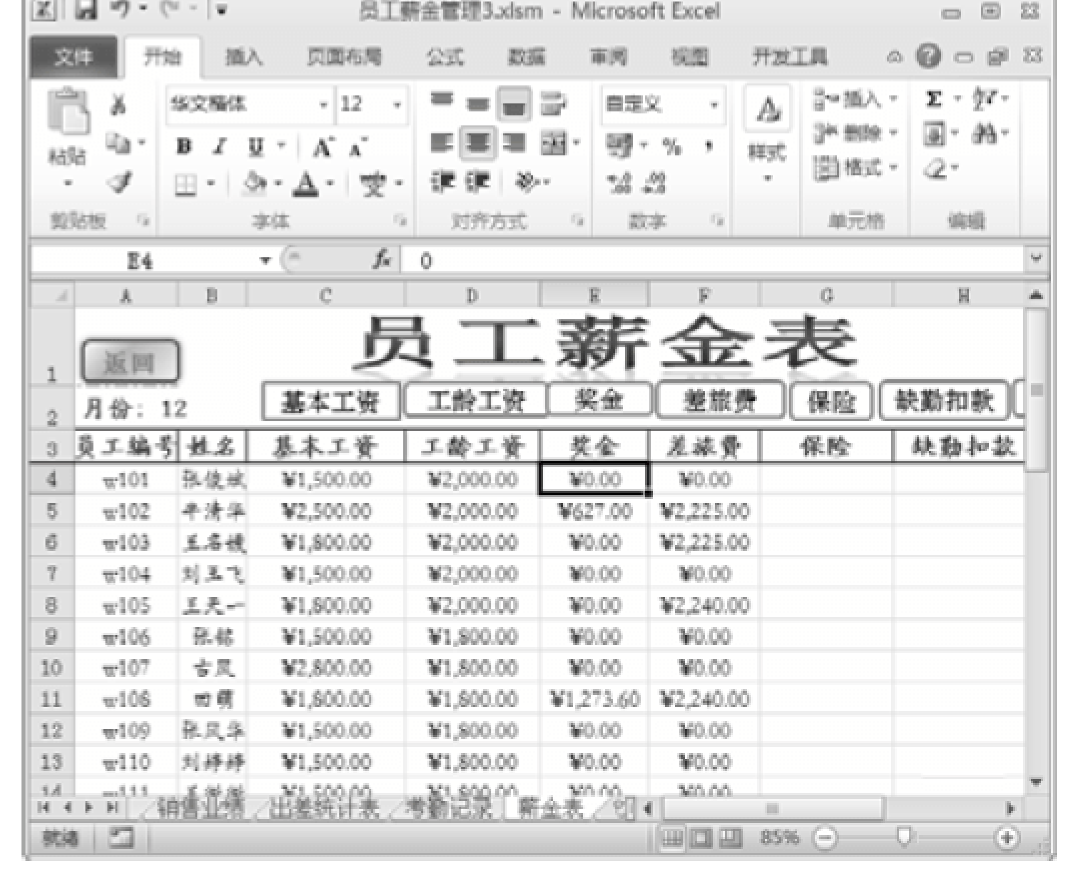


图 17-28 计算出每位员工的差旅费

07 上面介绍了“薪金表”工作表中前 4 种类型工资的管理过程，后面还有“保险”、“缺勤扣款”、“加班费”、“实发工资”4 个工资项，添加过程和前面的过程完全相同，只需要按照不同项目的计算公式进行计算就可以了。

为了演示操作过程，做如下假定：

- 每个员工的保险费用为固定值 150；
- 缺勤扣款计算方法为：“迟到或早退一次扣款 30 元，旷工一次罚款 200 元”；
- 加班费规定“加班一次支付给员工 100 元的加班费”；
- 实发工资=基本工资+工龄工资+奖金+差旅费-保险-缺勤扣款+加班费。

在 Excel 表格中按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 的代码编辑窗口中创建

计算后面几个工资项的子过程，输入代码如下。

```
Sub 保险()
    Dim row As Integer
    Dim count As Integer
    count = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count
       Cells(row, 7) = 150
    Next row
End Sub

Sub 缺勤扣款()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    Dim punish As Integer
    count1 = Sheets("考勤记录").Range("A1").CurrentRegion.Rows.count
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count1
       For col = 4 To count2
          If Cells(col, 1) = Sheets("考勤记录").Cells(row, 1) Then  '员工编号相
同
             punish = Sheets("考勤记录").Cells(row, 33) * 30 +
Sheets("考勤记录").Cells(row, 34) * 30 + Sheets("考勤记录").Cells(row, 35) * 200
             Cells(col, 8) = punish
             Exit For
          End If
       Next col
    Next row
End Sub

Sub 加班费()
    Dim row As Integer, col As Integer
    Dim count1 As Integer, count2 As Integer
    Dim overtime_pay As Integer
    count1 = Sheets("考勤记录").Range("A1").CurrentRegion.Rows.count
    count2 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    For row = 4 To count1
       For col = 4 To count2
          If Cells(col, 1) = Sheets("考勤记录").Cells(row, 1) Then  '员工编号相
同
             overtime_pay = Sheets("考勤记录").Cells(row, 36) * 100
             Cells(col, 9) = overtime_pay
             Exit For
          End If
       Next col
    Next row
End Sub

Sub 实发工资()
    Dim row As Integer
```

```
        Dim count As Integer
        count = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
        For row = 4 To count
            Cells(row, 10) = Cells(row, 3) + Cells(row, 4) + Cells(row, 5) + Cells(row,
6) - Cells(row, 7) - Cells(row, 8) + Cells(row, 9)
        Next row
    End Sub
```

08 输入完毕，返回“薪金表”工作表中，分别右击各个对应名称的圆角矩形，在弹出的快捷菜单中选择【指定宏】命令，再在打开的【指定宏】对话框中分别为每个图形按钮指定对应的操作宏（可参考前面的指定宏过程）。

添加完所有的操作宏之后，这些圆角矩形可以作为按钮来执行相应的 VBA 过程代码了。依次执行每个按钮，最后员工“薪金表”工作表的显示结果如图 17-29 所示。



员工编号	姓名	基本工资	工龄工资	奖金	差旅费	保险	缺勤扣款	加班费	实发工资
w101	张俊斌	¥1,500.00	¥2,000.00	¥0.00	¥0.00	¥150.00	¥30.00	¥200.00	¥3,520.00
w102	牛清华	¥2,500.00	¥2,000.00	¥627.00	¥2,225.00	¥150.00	¥30.00	¥100.00	¥7,272.00
w103	王名援	¥1,800.00	¥2,000.00	¥0.00	¥2,225.00	¥150.00	¥0.00	¥100.00	¥5,975.00
w104	刘玉飞	¥1,500.00	¥2,000.00	¥0.00	¥0.00	¥150.00	¥0.00	¥100.00	¥3,450.00
w105	王天一	¥1,800.00	¥2,000.00	¥0.00	¥2,240.00	¥150.00	¥60.00	¥0.00	¥5,830.00
w106	张然	¥1,500.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥0.00	¥0.00	¥3,150.00
w107	古风	¥2,800.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥0.00	¥100.00	¥4,550.00
w108	田萌	¥1,800.00	¥1,800.00	¥1,273.60	¥2,240.00	¥150.00	¥0.00	¥0.00	¥6,963.60
w109	张凤华	¥1,500.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥0.00	¥300.00	¥3,450.00
w110	刘婷婷	¥1,500.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥200.00	¥0.00	¥2,950.00
w111	王微微	¥1,500.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥30.00	¥300.00	¥3,420.00
w112	张小明	¥1,800.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥60.00	¥0.00	¥3,390.00
w113	李明远	¥1,500.00	¥1,800.00	¥0.00	¥0.00	¥150.00	¥0.00	¥300.00	¥3,450.00

图 17-29　计算出每位员工的实发工资



17.2 创建员工薪金管理窗体

员工薪金管理表已经创建好了，下面将介绍如何创建员工查询系统。该查询系统中通过创建一个查询窗体，将员工的详细资料和各项薪金数据显示出来。在该实例中还添加了用户权限管理部分，即增加了一个管理员角色，只有拥有管理员密码的人才能修改数据表中的数据，而其他普通员工只能做查询操作。

17.2.1 创建【薪金查询】窗体

薪金查询窗口中使用“员工编号”来查询，以防止出现名字相同的用户查询到错误的信息。用户只要在薪金查询窗口中输入员工编号就可以查看员工编号所对应的员工信息，具体的操作步骤如下。

01 打开本章的〝员工薪金管理 4〞工作簿，按 Alt+F11 组合键打开 VBA 代码窗口，依次选择【插入】➤【用户窗体】菜单命令，插入一个名为 UserForm1 的用户窗体，如图 17-30 所示。

02 在该用户窗体的【属性】窗口中的 Caption 文本框中输入〝薪金查询〞，修改该用户窗体的显示标题，如图 17-31 所示。

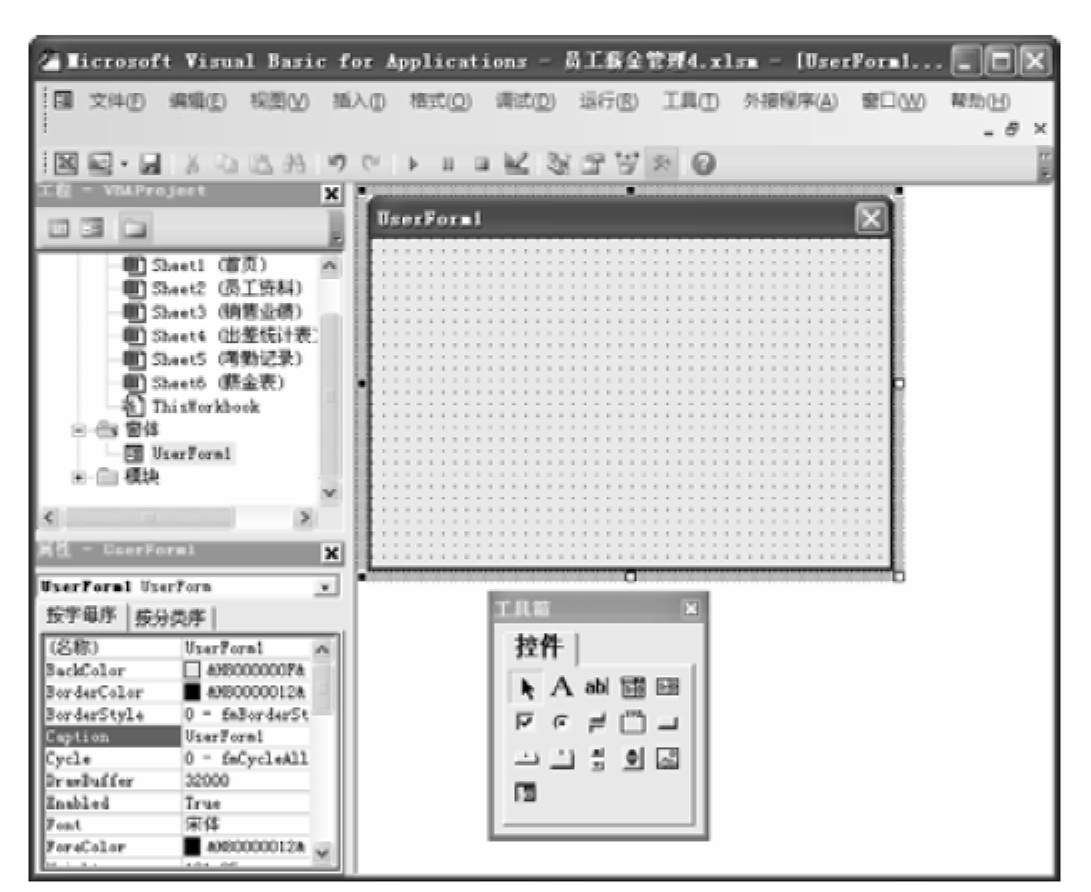


图 17-30　插入名为 UserForm1 的用户窗体

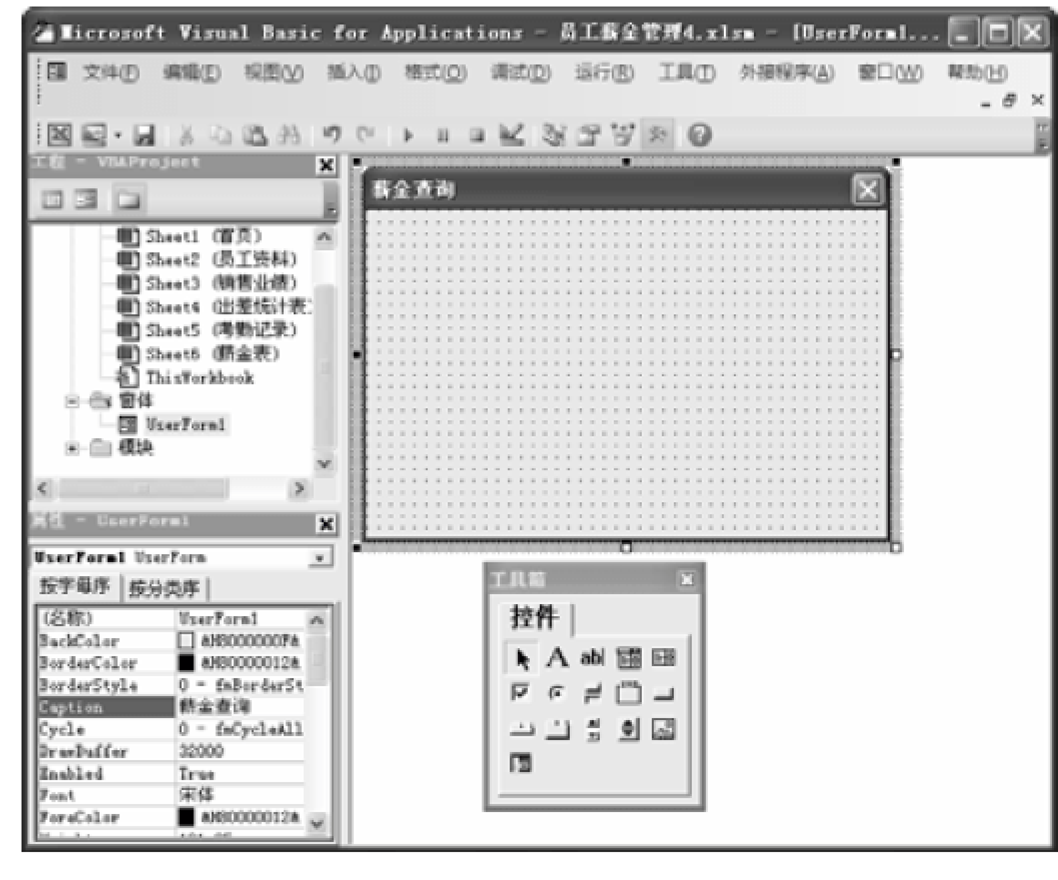


图 17-31　输入文本信息

03 添加提示标签。单击【工具箱】中的【标签】按钮A，在窗体上添加一个名为 Label1 的标签，修改该标签的 Caption 属性值为〝输入员工编号〞，修改结果如图 17-32 所示。

设置标签的显示文本字体格式，在该标签的【属性】窗口中单击 Font 属性右边带省略号的按钮，打开【字体】对话框，设置字体为〝华文中宋〞，大小为〝小四〞，如图 17-33 所示。

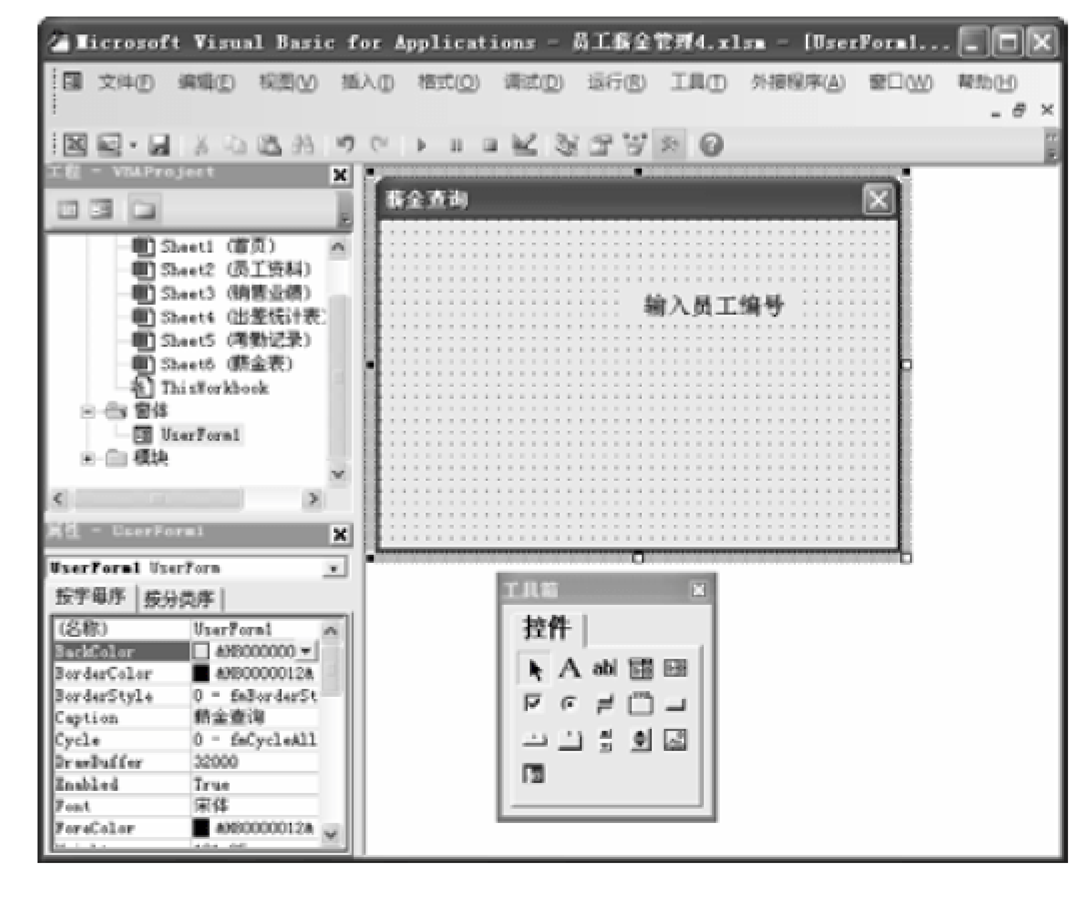


图 17-32　添加标签并输入标签文本



图 17-33　设置文本属性

单击【确定】按钮关闭【字体】对话框。然后单击【工具箱】中的【文本框】按钮abl，在窗体上添加一个名为 TextBox1 的文本框，如图 17-34 所示。

04 单击【工具箱】中的【命令按钮】按钮，在窗体中添加一个命令按钮，修改该命令按钮的 Caption 属性值为“开始查询”，输入完成后，按 Enter 键完成命令按钮显示文本的修改，如图 17-35 所示。

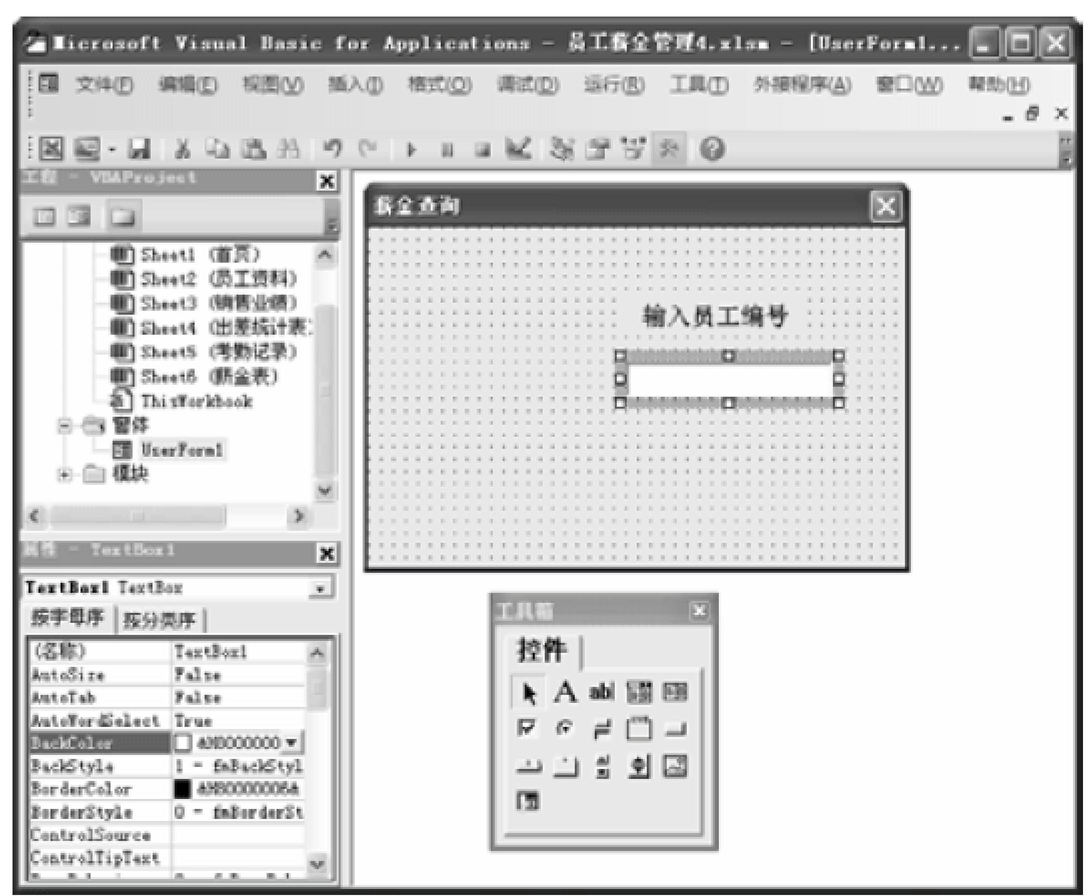


图 17-34　添加一个名为 TextBox1 的文本框



图 17-35　添加一个命令按钮

05 按照添加【开始查询】按钮的方法，在窗体中添加另外两个命令按钮，分别修改这两个命令按钮的 Caption 属性值为“管理员登录”和“退出”，如图 17-36 所示。

首先，双击用户窗体中【开始查询】按钮进入其代码编辑窗口，添加按钮的 Click 事件过程的处理代码。

```
Private Sub CommandButton1_Click()
    Dim row As Integer
    Dim count1, count2, count3, count4, count5 As Integer
    Dim flag1 As Boolean, flag2 As Boolean, flag3 As Boolean

    count1 = Sheets("员工资料").Range("A1").CurrentRegion.Rows.count
    count2 = Sheets("销售业绩").Range("C3").CurrentRegion.Rows.count + 2
    count3 = Sheets("出差统计表").Range("B3").CurrentRegion.Rows.count + 2
    count4 = Sheets("考勤记录").Range("A1").CurrentRegion.Rows.count
    count5 = Sheets("薪金表").Range("A1").CurrentRegion.Rows.count
    isTravel = False
    flag1 = False
    flag2 = False
    flag3 = False
    ID = TextBox1.Text  '记录用户输入的员工编号

    '获取员工个人资料
    For row = 3 To count1
       If Sheets("员工资料").Cells(row, 1) = ID Then
```




```
            Sname = Sheets("员工资料").Cells(row, 2)  '员工姓名
            degree = Sheets("员工资料").Cells(row, 4) '员工学历
            depart = Sheets("员工资料").Cells(row, 5) '员工所在部门
            work_time = Sheets("员工资料").Cells(row, 7) '工作时间
            flag1 = True  '说明此员工存在
            Exit For
          End If
       Next row
       '获取销售额
       For row = 4 To count2
          If Sheets("销售业绩").Cells(row, 3) = ID Then
            sale = Sheets("销售业绩").Cells(row, 5)  '销售额
            Exit For
         End If
       Next row
       '获取出差日期
       For row = 4 To count3
         If Sheets("出差统计表").Cells(row, 2) = ID Then
           travel = Sheets("出差统计表").Cells(row, 7) '出差天数
           isTravel = True
           Exit For
         End If
       Next row
       '获取考勤信息
       For row = 4 To count4
         If Sheets("考勤记录").Cells(row, 1) = ID Then
           late = Sheets("考勤记录").Cells(row, 33)  '迟到次数
           early = Sheets("考勤记录").Cells(row, 34) '早退次数
           truancy = Sheets("考勤记录").Cells(row, 35) '旷工次数
           overtime = Sheets("考勤记录").Cells(row, 36) '加班次数
           flag2 = True '说明此员工存在
           Exit For
         End If
       Next row
       '获取各工资项目并求总工资
       For row = 4 To count5
         If Sheets("薪金表").Cells(row, 1) = ID Then
           basic_money = Sheets("薪金表").Cells(row, 3) '基本工资
           time_money = Sheets("薪金表").Cells(row, 4)  '工龄工资
           prize money = Sheets("薪金表").Cells(row, 5) '奖金
           travel money = Sheets("薪金表").Cells(row, 6) '差旅费
           punish money = Sheets("薪金表").Cells(row, 8) '缺勤扣款
           overwork money = Sheets("薪金表").Cells(row, 9) '加班费
           salary = Sheets("薪金表").Cells(row, 10) '实发工资
           flag3 = True '说明此员工存在
           Exit For
         End If
       Next row

       If flag1 = False Or flag2 = False Or flag3 = False Then
```

```
            MsgBox "不存在这个员工编号！"
        Else
            UserForm1.Hide
            UserForm2.Show
        End If
    End Sub
```

这段代码需要从相关的工作表中查找与所输入员工编号对应的员工详细信息。这里使用 For 循环遍历“员工资料”、“销售业绩”、“出差统计表”、“考勤记录”和“薪金表”这 5 个工作表。

如果在工作表中存在所输入的员工编号，则使用 UserForm1.Hide 语句隐藏【员工薪金查询】窗体，然后使用 UserForm2.Show 语句将 UserForm2 查询结果窗体显示出来。

然后，双击【薪金查询】窗体中的【管理员登录】按钮，进入该按钮的代码编辑窗口，添加按钮的 Click 事件过程的处理代码。

```
    Private Sub CommandButton2_Click()
        UserForm1.Hide   '隐藏【薪金查询】窗体
        UserForm3.Show   '显示【管理员登录】窗体
    End Sub
```

【提示】这里的 UserForm3 用户窗体（即【管理员登录】窗体），后面的小节中将创建该窗体。

最后，双击【薪金查询】窗体中的【退出】按钮，进入该按钮的代码编辑窗口，添加按钮的 Click 事件过程的处理代码。

```
    Private Sub CommandButton3_Click()
        UserForm1.Hide '隐藏【薪金查询】窗体
        ActiveWorkbook.Close (True)  '保存并关闭工作簿
    End Sub
```

【提示】在上述 CommandButton1_Click 事件处理程序中使用到的一些保存员工信息的变量并没有定义，而这些变量会在该工作簿的其他窗体中使用到，即这些变量是公共的，所以接下来的步骤将把这些变量定义成全局变量。

06 在 VBA 代码窗口中，选择【插入】➢【模块】菜单命令，插入〝模块 2〞。在该模块的【属性】窗口中的【名称】文本框中输入〝定义变量〞，接着在该模块代码窗口中输入定义变量的代码，如图 17-37 所示。

```
    Public ID As String, Sname As String
    '保存员工学历变量
    Public degree As String, depart As String, work_time As String
    '保存迟到、早退、旷工变量
    Public late As Integer, early As Integer, truancy As Integer
    '加班变量
    Public overtime As Integer
    '销售额、奖金变量
    Public sale As Double, prize_money As Double
```




```
'基本工资、工龄工资
Public basic money As Integer, time money As Integer
'扣款、加班费变量
Public punish_money As Integer, overwork_money As Integer
'出差天数、差旅费
Public travel As Integer, travel_money As Integer
'总工资
Public salary As Double
'员工出差判断变量
Public isTravel As Boolean
```



图 17-36　添加【管理员登录】和【退出】按钮



图 17-37　定义变量代码

07 最后必须添加一个用户窗体关闭事件的处理过程，因为在查询系统中普通用户不能查看 Excel 表格中的数据，所以当用户关闭窗体时，一定要保证把 Excel 工作簿关闭。下面继续输入如下代码以实现关闭用户窗体时自动关闭工作簿的功能。

```
Private Sub UserForm_Terminate()
ActiveWorkbook.Close (False)
End Sub
```

17.2.2 创建【查询结果】窗体

现在已经完成了【薪金查询】窗体的创建过程，查询之后的结果还需要一个窗体来显示，所以接下来将介绍显示【查询结果】的用户窗体的创建和设计过程。该窗体将显示相应员工的个人薪金信息，具体的操作步骤如下。

01 在 VBA 代码窗口中依次选择【插入】➢【用户窗体】菜单命令，插入一个名为 UserForm2 的用户窗体，将其名称属性值修改为 "查询结果"，如图 17-38 所示。

02 按照前面介绍的添加标签的方法，在 UserForm2 用户窗体中分别添加如图 17-39 所示的标签。

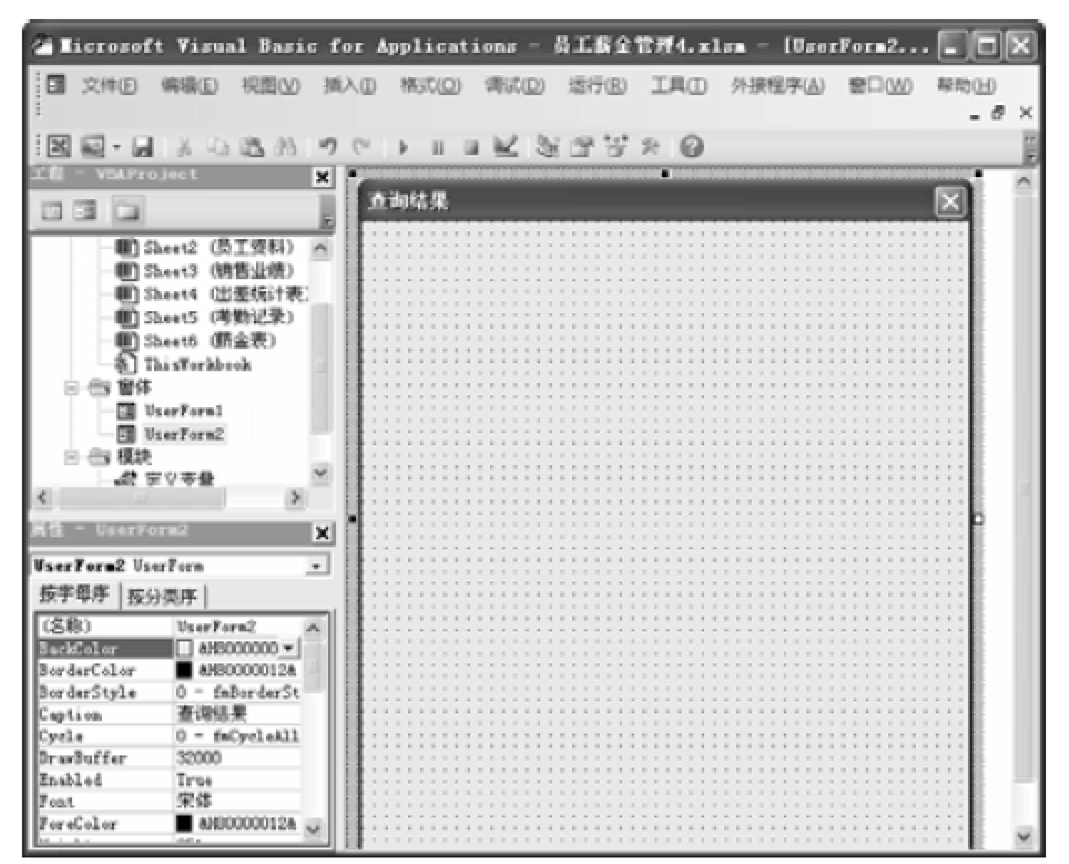


图 17-38　修改窗体显示标题



图 17-39　添加多个标签

03 修改各个标签的"名称"属性。选中【销售额】标签，修改其"名称"属性值为 Label_Sale，如图 17-40 所示。

04 按照同样的方法将【出差】标签的名称修改为 Label_Out，将【天】标签的名称修改为 Label_Day，将【差旅费】标签的名称修改为 Label_Outpay；将【奖金】标签的名称修改为 Label_Prize，如图 17-41 所示。

05 按照前面添加文本框的方法，在 UserForm2 用户窗体中分别添加如图 17-42 所示的文本框。

06 按照统一的命名格式对文本框进行重命名。例如，选择【编号】文本框，修改其"名称"属性值为 Txt_ID，使该文本框与定义的全局变量 ID 相对应。按照这种方法分别将其他文本框进行重命名，各个文本框的名称对应如下：

- "姓名"文本框的名称修改为 Txt_Name；
- "部门"文本框的名称修改为 Txt_Depart；
- "学历"文本框的名称修改为 Txt_Degree；
- "参加工作时间"文本框的名称修改为 Work_Date；
- "基本工资"文本框的名称修改为 Basic_Pay；
- "工龄工资"文本框的名称修改为 WorkDate_pay；
- "迟到次数"文本框的名称修改为 Txt_Late；
- "早退次数"文本框的名称修改为 Txt_Early；
- "旷工次数"文本框的名称修改为 Txt_Truancy；
- "缺勤扣款"文本框的名称修改为 Txt_Punish；
- "加班次数"文本框的名称修改为 Over_Work；
- "加班费"文本框的名称修改为 Overwork_Pay；
- "出差天数"文本框的名称修改为 Txt_Travel；
- "差旅费"文本框的名称修改为 Travel_pay；
- "销售额"文本框的名称修改为 Txt_Sale；
- "奖金"文本框的名称修改为 Sale_Pay；
- "本月实发工资"文本框的名称修改为 Txt_money。

这样所有的文本框名称将分别对应各自的全局变量。

07 按照前面添加命令按钮的方法，在该用户窗体中分别添加两个显示文本为＂返回＂和＂退出系统＂的命令按钮，如图 17-43 所示。

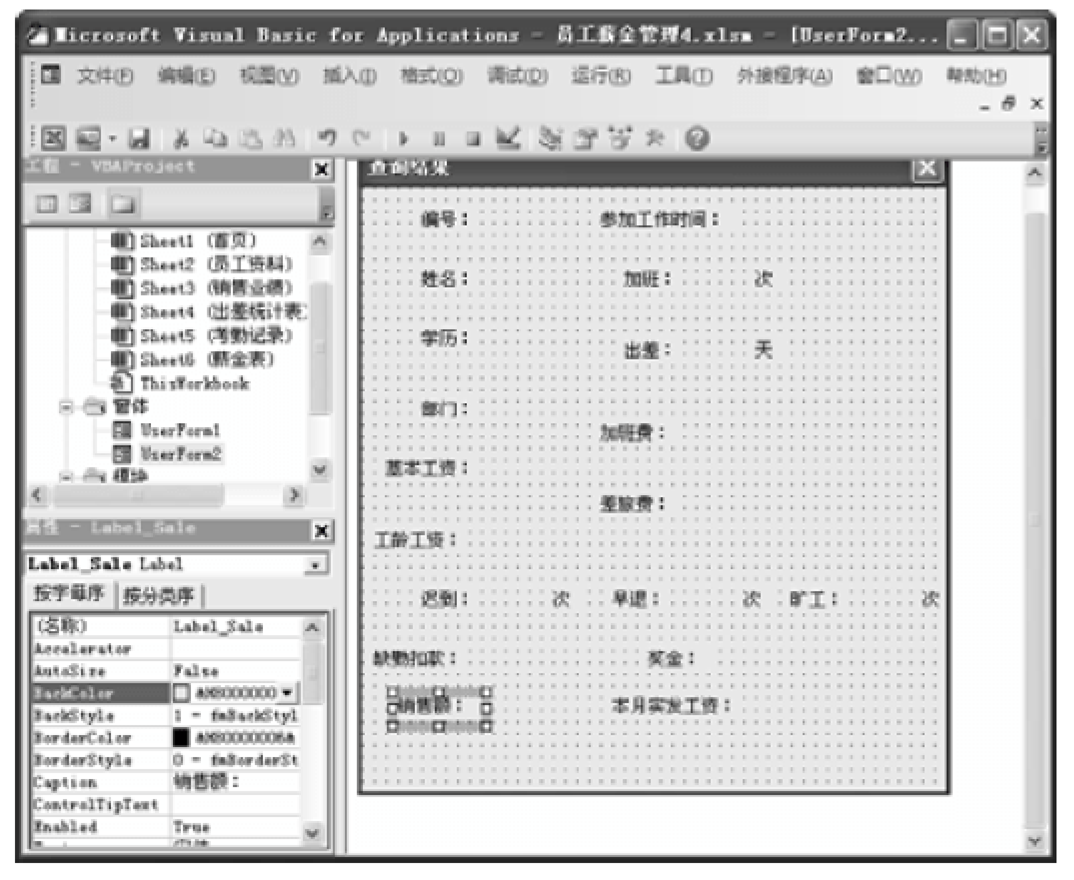


图 17-40　修改【销售额】标签的名称属性



图 17-41　修改【奖金】标签的名称属性



图 17-42　添加多个文本框



图 17-43　添加两个命令按钮

在【查询结果】窗体中，显示所查询的员工的详细信息，需要对 UserForm2 用户窗体的 Activate 事件进行处理。在此事件处理程序中将查询到的员工详细信息显示在 UserForm2 用户窗体上相应的文本框中。

08 在【工程资源管理器】窗口中右击 UserForm2 选项，在弹出的快捷菜单中选择【查看代码】菜单命令，或者在 UserForm2 窗体中双击鼠标，打开 UserForm2 用户窗体的代码编辑窗口，添加 Activate 事件的处理代码。

```
Private Sub UserForm_Activate()
    Txt_ID.Text = ID
    Txt_Name.Text = Sname
```

```
        Txt Depart.Text = depart
        Txt Degree.Text = degree
        Basic_Pay.Text = basic_money
        Work_Date.Text = work_time
        WorkDate_pay.Text = time_money
        Txt_Late.Text = late
        Txt_Early.Text = early
        Txt_Truancy.Text = truancy
        Txt_Punish.Text = punish_money
        Over_Work.Text = overtime
        Overwork_Pay.Text = overwork_money
        '如果员工出差，则显示关于出差天数和差旅费的标签与文本框
        If flag = True Then
           Label Out.Visible = True
           Label Outpay.Visible = True
           Txt Travel.Visible = True
           Travel_pay.Visible = True
           Label_Day.Visible = True
           Txt_Travel.Text = travel
           Travel_pay.Text = travel_money
        Else
           Label_Day.Visible = False
           Label_Out.Visible = False
           Label_Outpay.Visible = False
           Txt_Travel.Visible = False
           Travel pay.Visible = False
        End If
        '如果是销售部的员工，则显示关于销售额和奖金的标签与文本框
        If depart = "销售部" Then
           Label_Sale.Visible = True
           Label_Prize.Visible = True
           Txt_Sale.Visible = True
           Sale_Pay.Visible = True
           Txt_Sale.Text = sale
           Sale_Pay.Text = prize_money
        Else
           Label_Sale.Visible = False
           Label_Prize.Visible = False
           Txt_Sale.Visible = False
           Sale Pay.Visible = False
        End If
        Txt money.Text = pay
    End Sub
```

09 接下来，分别为【查询结果】窗体中的各个按钮添加 Click 事件的处理代码。首先双击【返回】按钮，进入该按钮的代码编辑窗口，输入代码如下。

```
    Private Sub CommandButton1 Click()
        UserForm2.Hide       '隐藏【查询结果】窗体
        UserForm1.Show       '显示【薪金查询】窗体
```




```
End Sub
```

然后，双击【退出系统】按钮，进入该按钮的代码编辑窗口，输入代码如下。

```
Private Sub CommandButton2_Click()
    UserForm2.Hide              '隐藏【查询结果】窗体
    ActiveWorkbook.Close (True) '保存并关闭工作簿
End Sub
```

10 最后，与【薪金查询】窗体相同，必须添加一个用户窗体关闭事件的处理过程，以保证当用户关闭窗口时关闭 Excel 工作簿，在代码窗口中输入下面的语句。

```
Private Sub UserForm_Terminate()
    ActiveWorkbook.Close (False)
End Sub
```

到此就完成了【查询结果】用户窗体的代码编写。

17.2.3 创建【管理员登录】窗体

最后将介绍员工薪金管理系统中的权限管理部分。权限管理可以保护管理系统中的数据，防止无关人员对工作簿进行修改。这里创建一个【管理员登录】窗体，只有拥有管理员密码的用户才能进入工作簿，具体的操作步骤如下。

01 在 VBA 代码窗口中选择【插入】➤【用户窗体】菜单命令，插入一个名为 UserForm3 的用户窗体，然后将其显示标题修改为"管理员登录"，如图 17-44 所示。

02 单击【工具箱】中的【标签】按钮 A，在窗体中添加一个名为 Label1 的标签，然后修改该标签的 Caption 属性值为"输入密码："，如图 17-45 所示。

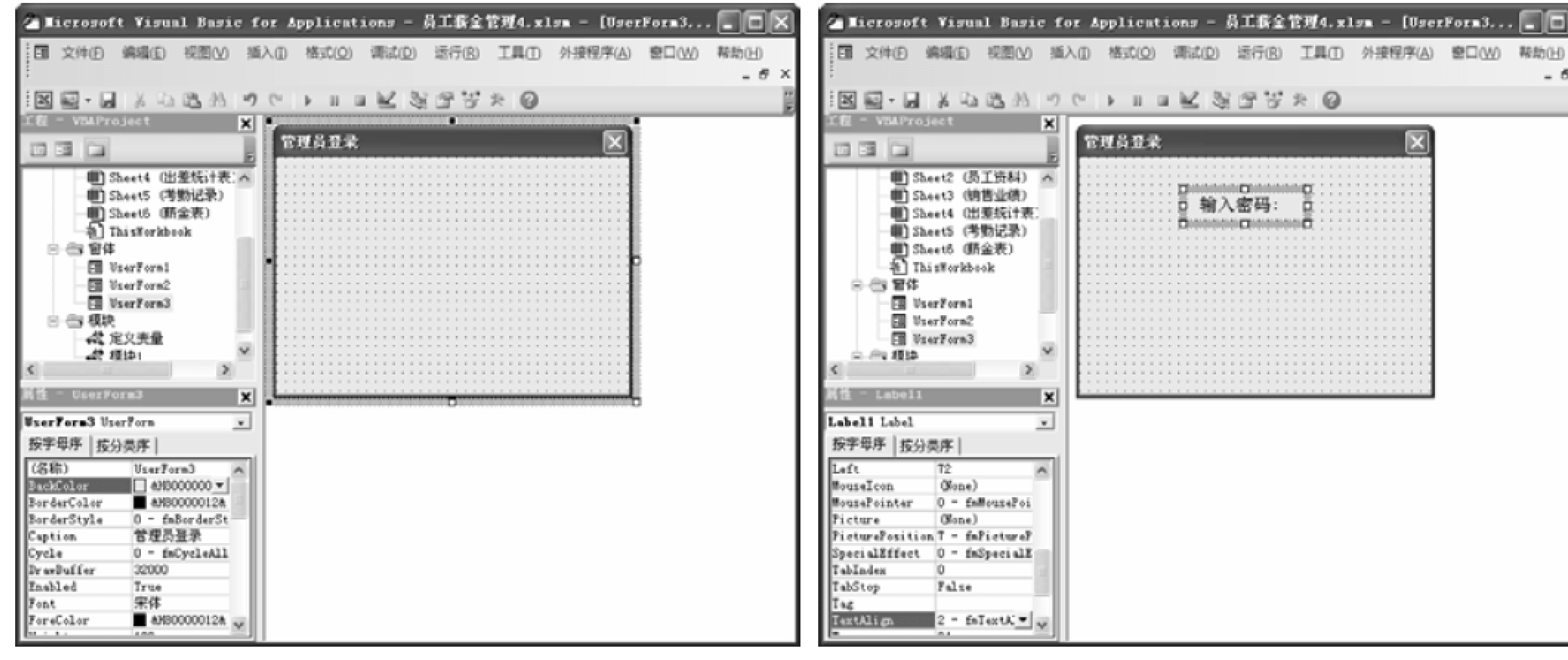


图 17-44　插入 UserForm3 用户窗体



图 17-45　添加标签并修改其显示文本

单击 Font 属性中带省略号的按钮，打开【字体】对话框，设置该标签的字体样式，如图 17-46 所示，设置完成后单击【确定】按钮完成对该标签字体的设置。

03 单击【工具箱】中的【文本框】按钮，在窗体中添加一个名为 TextBox1 的文本框，如图 17-47 所示。

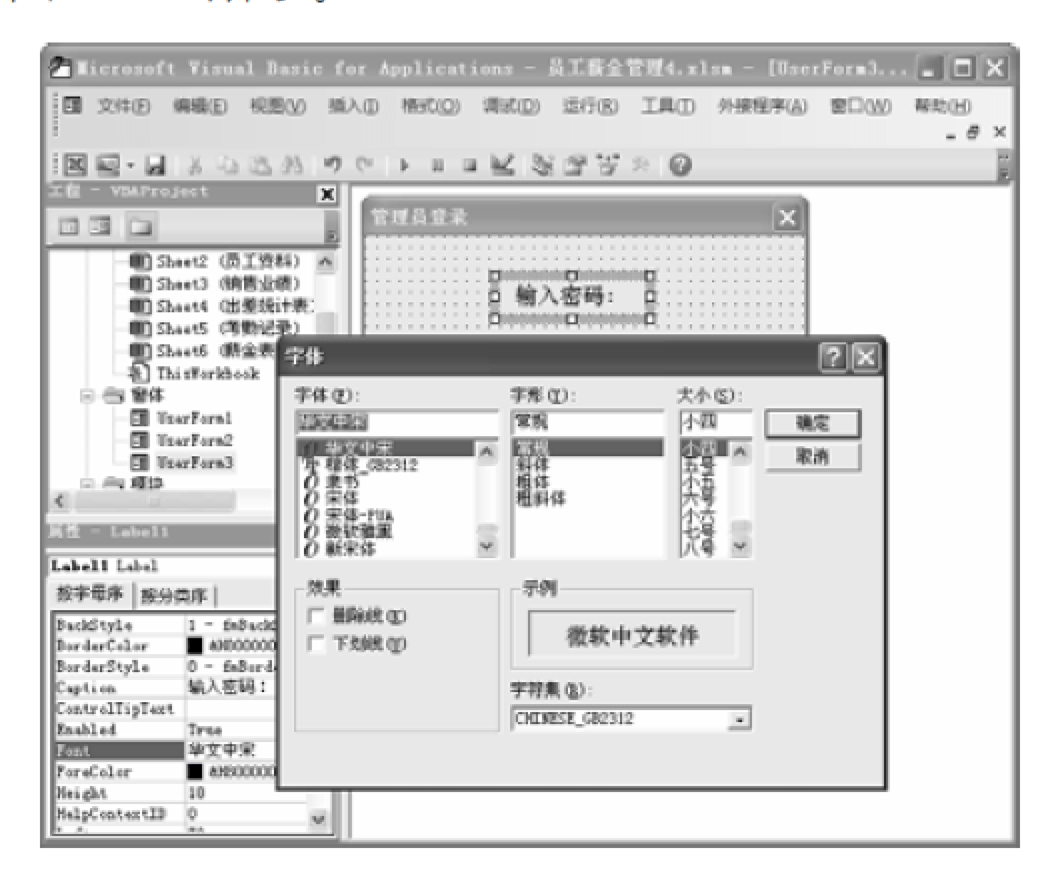


图 17-46　设置标签的字体



图 17-47　添加文本框

修改该文本框 PasswordChar 属性值为“*”。之后在使用 TextBox1 文本框时，其中输入的密码都会显示为相应长度的星号，如图 17-48 所示。

04 单击【工具箱】中的【命令按钮】按钮，在窗体中添加两个命令按钮，并将其 Caption 属性分别设置为“登录”和“返回”，如图 17-49 所示。



图 17-48　设置 PasswordChar 的属性值



图 17-49　添加两个命令按钮

接下来为这两个按钮指定 Click 事件。首先双击【登录】按钮，进入该按钮的代码编辑窗口，输入下面的代码程序。

```
Private Sub CommandButton1_Click()
    If TextBox1.Text = "123456" Then
       UserForm3.Hide         '隐藏【管理员登录】窗体
    Else
       MsgBox "密码不正确！"
    End If
```




```
End Sub
```

这段代码实际上是设置了管理员登录密码。在实例中将密码设置为“123456”，如果用户输入此密码，则可隐藏【管理员登录】窗体，进入工作簿中；如果密码输入不正确，则系统会自动弹出提示信息。

然后双击【返回】按钮，进入该按钮的代码编辑窗口，输入下面的程序代码：

```
Private Sub CommandButton2_Click()
    UserForm3.Hide  '隐藏【管理员登录】窗体
    UserForm1.Show  '显示【薪金查询】窗体
End Sub
```

05 为了避免在关闭【管理员登录】窗体时进入工作簿，需要对此窗体的关闭事件进行处理，即在上述代码窗口中输入下面的程序语句。

```
Private Sub UserForm_Terminate()
    ActiveWorkbook.Close (False)
End Sub
```

06 为了确保用户在打开工作簿时即可进入【薪金查询】窗体，而不是直接进入工作簿，需要设置工作簿的打开事件。双击【工程资源管理器】窗口中的 ThisWorkBook 选项，进入其代码编辑窗口，输入下面的代码程序。

```
Private Sub Workbook_Open()
    Sheets("首页").Activate  '显示工作簿的首页
    UserForm1.Show   '显示【薪金查询】窗体
End Sub
```

07 返回“员工薪金管理 4”工作簿中，切换到“首页”工作表，选择【文件】→【选项】菜单命令，即可打开【Excel 选项】对话框，在【高级】选项卡中，取消选中【显示水平滚动条】、【显示工作表标签】和【显示行和列标题】复选框，如图 17-50 所示。单击【确定】按钮保存设置并返回“首页”工作表中，其效果如图 17-51 所示。

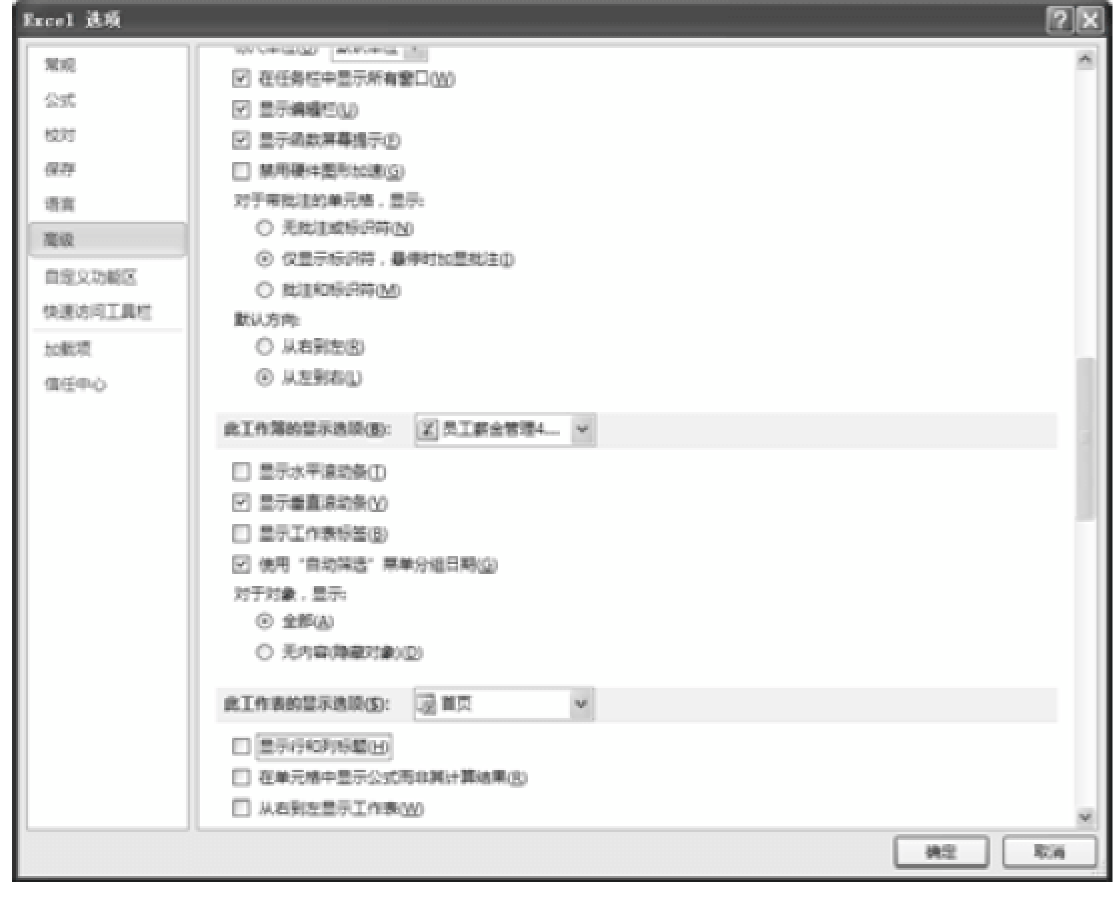


图 17-50 【Excel 选项】对话框

图 17-51 隐藏各项信息后的效果



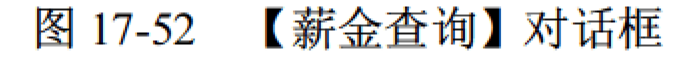
17.3 运行员工薪金管理系统

至此已经完成薪金管理系统的创建了，接下来，体验一下自己设计的系统吧。

01 重新打开“员工薪金管理 4”工作簿，此时，系统首先将打开【薪金查询】对话框，输入查询的员工编号，就可以获得某个员工的个人信息，例如输入 w116，如图 17-52 所示。

02 输入编号之后，单击【开始查询】按钮，系统将自动关闭【薪金查询】对话框，并显示【查询结果】对话框，从中可以查看该员工的详细信息，如图 17-53 所示。




图 17-52　【薪金查询】对话框

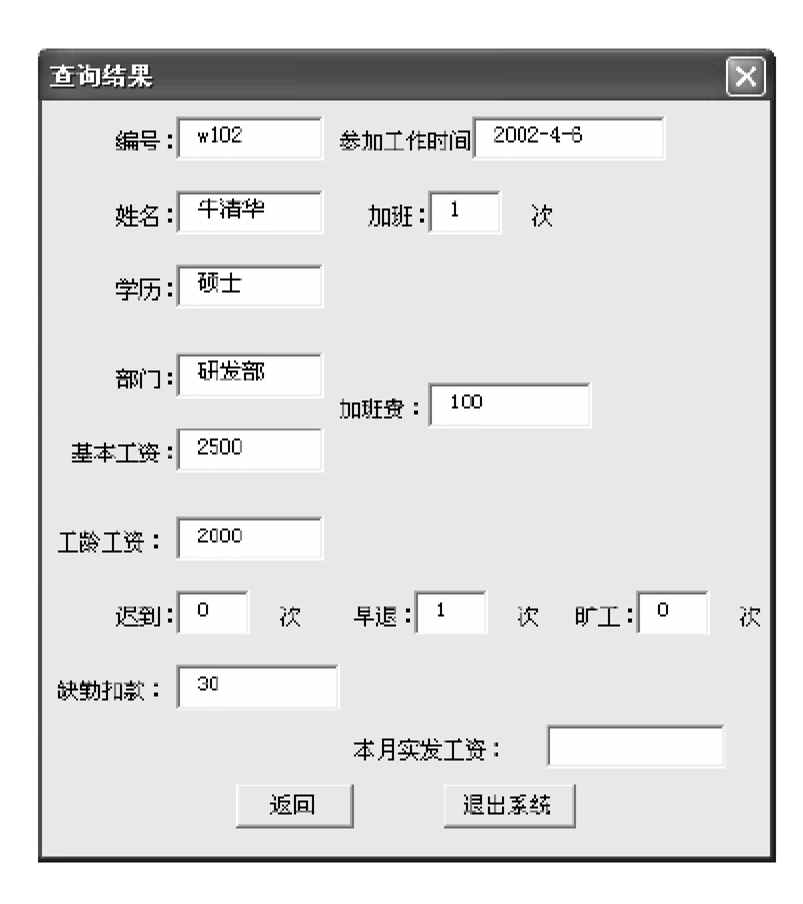


图 17-53　【查询结果】对话框

03 单击【返回】按钮即可关闭【查询结果】对话框，并返回到【薪金查询】对话框中，在文本框中输入另一位员工的编号，这里输入 w101，如图 17-54 所示。

04 单击【开始查询】按钮，进入【查询结果】对话框，可以看到这次的查询结果中没有显示销售额和奖金信息，因为该员工不是销售部员工，所以该员工会有出差记录，也不会显示出差的信息，如图 17-55 所示。




图 17-54　输入员工编号 w101

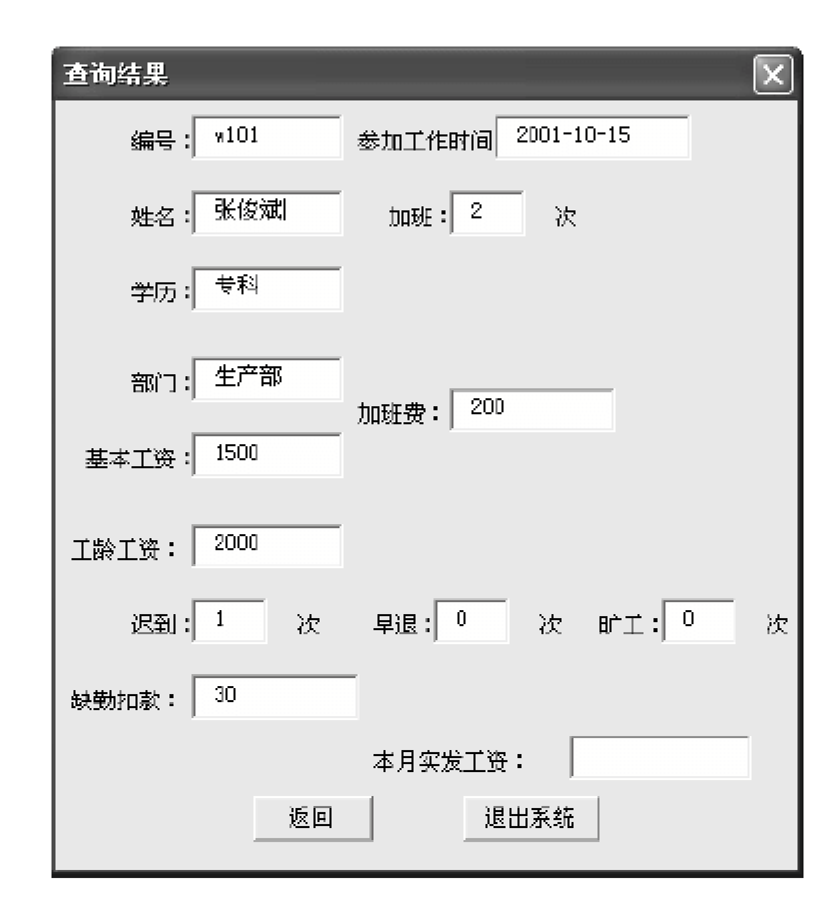


图 17-55　显示查询结果

05 单击【返回】按钮返回到【薪金查询】对话框中。输入 w220，然后单击【开始查询】按钮，此时系统会弹出"不存在这个员工编号！"的提示信息，这是因为输入了一个在工作簿中不存在的员工编号，如图 17-56 所示。

06 单击提示信息对话框中的【确定】按钮，返回到【薪金查询】对话框中，如果用户拥有管理员密码，就可以单击【管理员登录】按钮，进入【管理员登录】对话框中，如图 17-57 所示。



图 17-56　员工编号不存在的提示信息

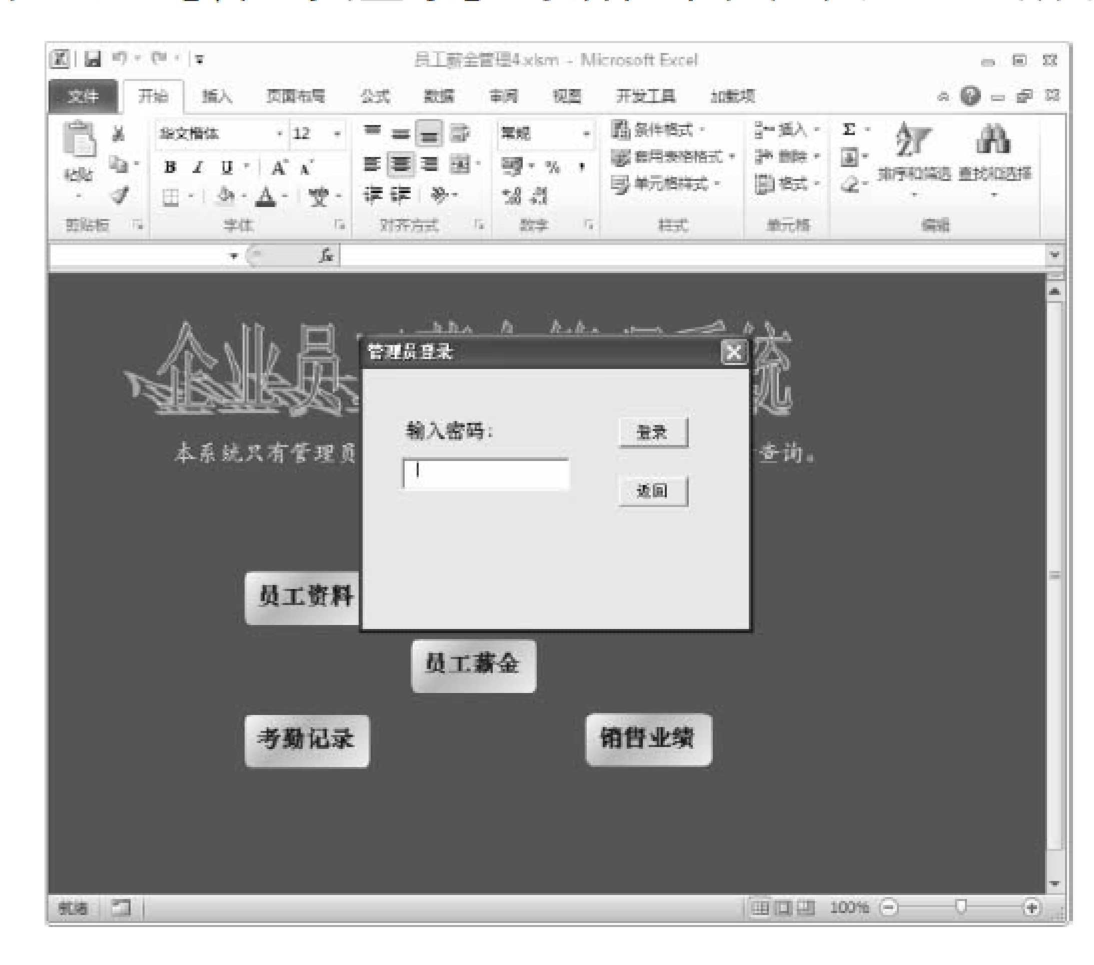


图 17-57　【管理员登录】对话框

07 在【管理员登录】对话框中输入正确的密码"123456"，然后单击【登录】按钮，系统就会自动关闭【管理员登录】对话框，并进入"员工薪金管理 4.xlsm"工作簿中。

08 如果在该对话框中的【密码】文本框中输入不正确的密码，例如输入"abcdef"，然后单击【登录】按钮，系统就会弹出"密码不正确！"的提示信息，如图 17-58 所示。



图 17-58　密码错误提示信息

管理员进入员工薪金管理系统后就可以进行各种操作了，如进行前面所讲述的汇总考勤记录或计算员工薪金的操作，还可以进行添加新员工、修改员工信息、完善程序代码等操作。

第 18 章　企业收支管理系统

企业的收支情况决定了企业的利润。通过建立收支管理系统，企业可以对各种收入和支出进行录入管理，方便财务核算企业收入项目，帮助公司进行财务规划。使用 Excel VBA 创建企业收支管理系统，可以极大地方便企业对收支情况进行统计、汇总、分析和预测。



18.1 设计“企业收支管理”工作簿

本节将创建的“企业收支管理”工作簿包含 6 个工作表，分别是收入登记、支出登记、收入汇总、支出汇总、收支对比和收支项目。

18.1.1 设计收支登记工作表

收支登记工作表包含了收入登记工作表和支出登记工作表，这两个工作表一个用于登记收入，一个用于登记支出，但是其中的内容是相同的。下面将介绍创建收支登记工作表的具体操作步骤。

01 首先创建一个 Excel 工作簿，将 Sheet1 工作表的名称修改为〝收入登记〞，然后在该工作表中输入下面的一些标题文字并修改标题的背景色，如图 18-1 所示；切换到【插入】选项卡，在【插图】选项组中的【形状】下拉列表中单击【矩形】组中的【圆角矩形】按钮▢，在第一行中绘制一个圆角矩形按钮，如图 18-2 所示。

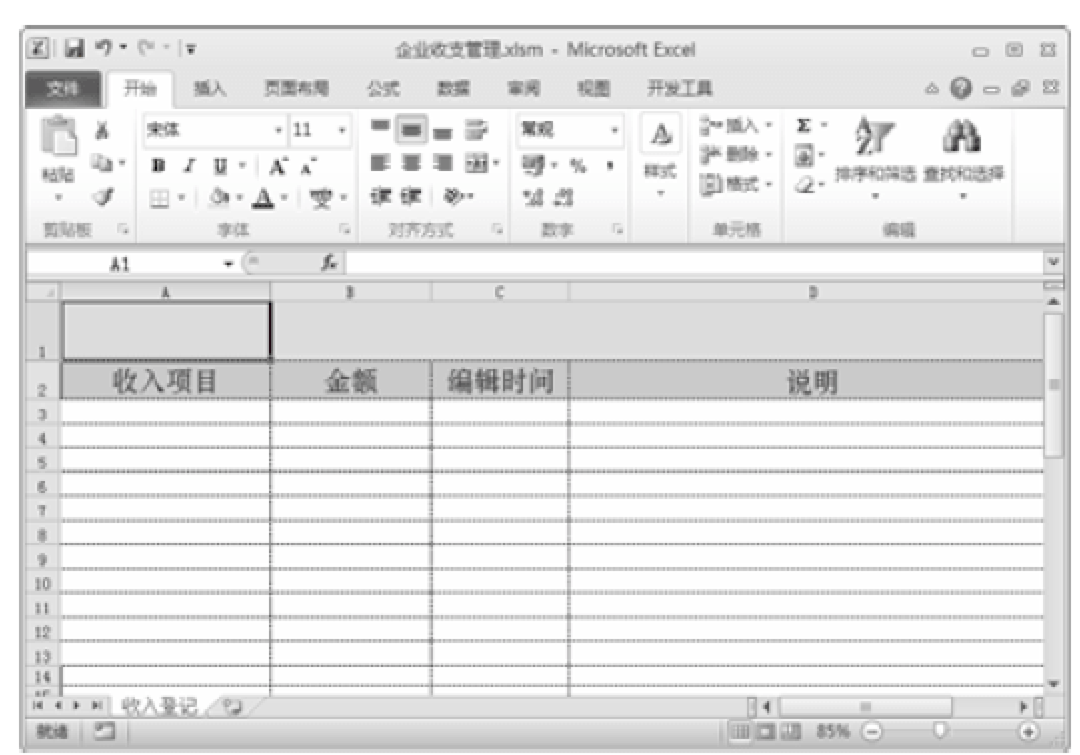


图 18-1　“收入登记”工作表

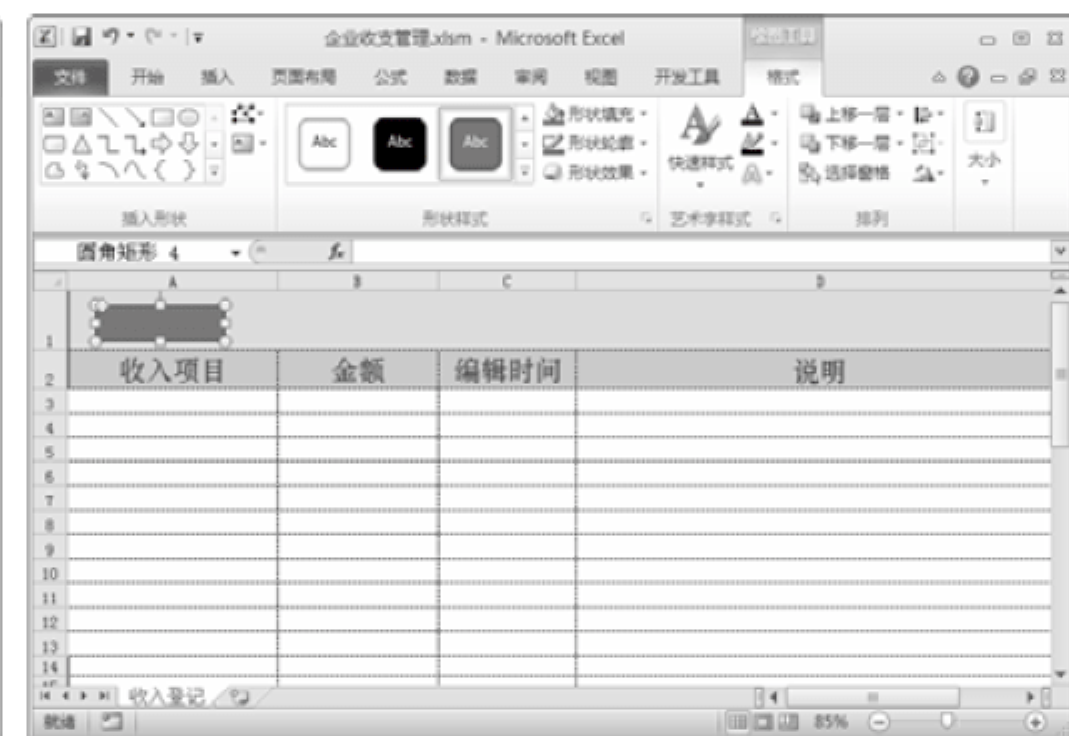


图 18-2　绘制一个圆角矩形按钮

单击【格式】选项卡【形状样式】选项组中的【其他】按钮，选择下拉列表中的形状

样式，如图 18-3 所示。

右击添加的圆角矩形，选择弹出菜单中的【编辑文字】菜单命令，添加“收入登记”文本，然后分别设置文本的“字体”为“华文中宋”，字号为“12”，对齐方式为“居中”，如图 18-4 所示。



图 18-3　设置按钮样式　　　　图 18-4　添加文本内容

02 按照上述添加圆角矩形的方法在工作表中添加其他几个圆角矩形，如图 18-5 所示。

03 插入艺术字。在【插入】选项卡中，单击【文本】选项组中的【艺术字】按钮，即可弹出各种艺术字效果，如图 18-6 所示。

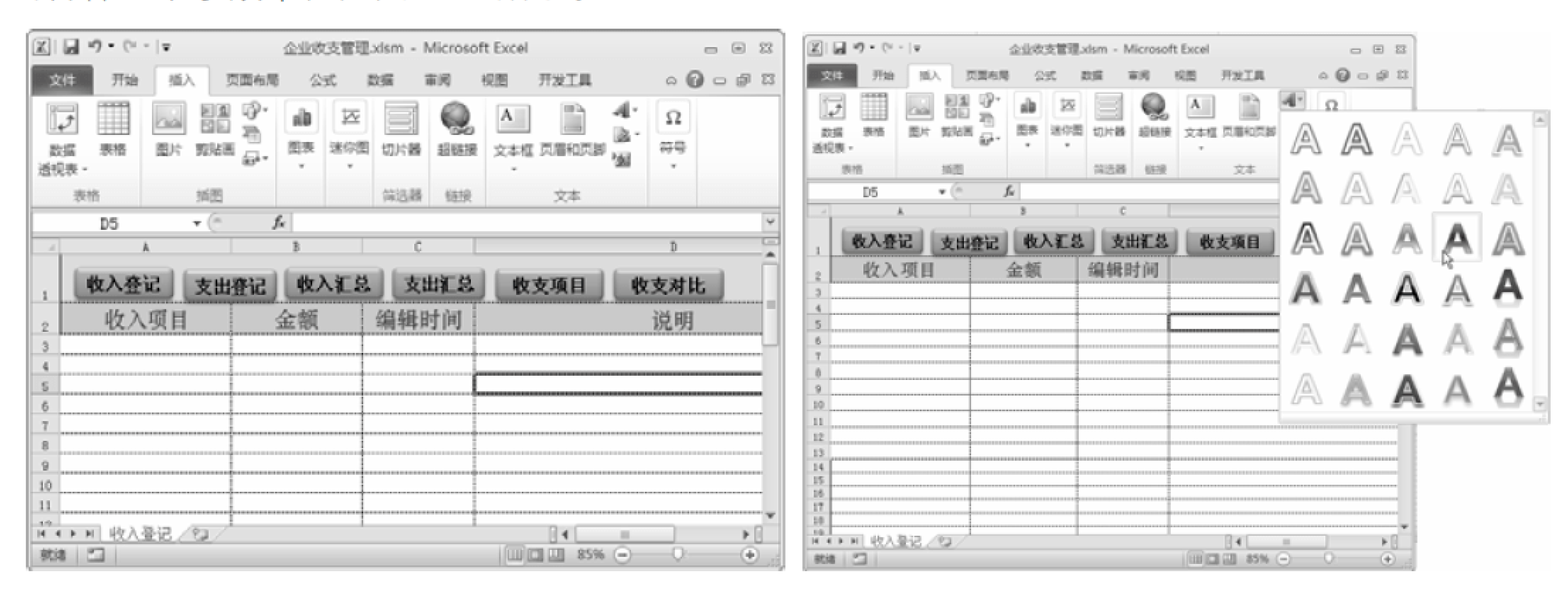


图 18-5　添加其他几个圆角矩形　　　　图 18-6　选择艺术字效果

从中选择相应的艺术字效果，即可弹出【请在此放置您的文字】标签，如图 18-7 所示。从中输入文本内容“登记”，并设置其字体、字号，然后将其放置在工作表的右上角，如图 18-8 所示。





图 18-7 【请在此放置您的文字】标签　　　　图 18-8 设置文本内容

04 在“收入登记”工作表中按下 Ctrl+A 组合键选中表中的全部内容，再按下 Ctrl+C 组合键复制工作表中的内容，然后切换到第二个工作表中，按下 Ctrl+V 组合键粘贴复制的内容。将第二个工作表的标签修改为“支出登记”，再将表中的 A2 单元格的内容修改为“支出项目”，其他内容保持不变。这样“支出登记”工作表也设置完成了，如图 18-9 所示。

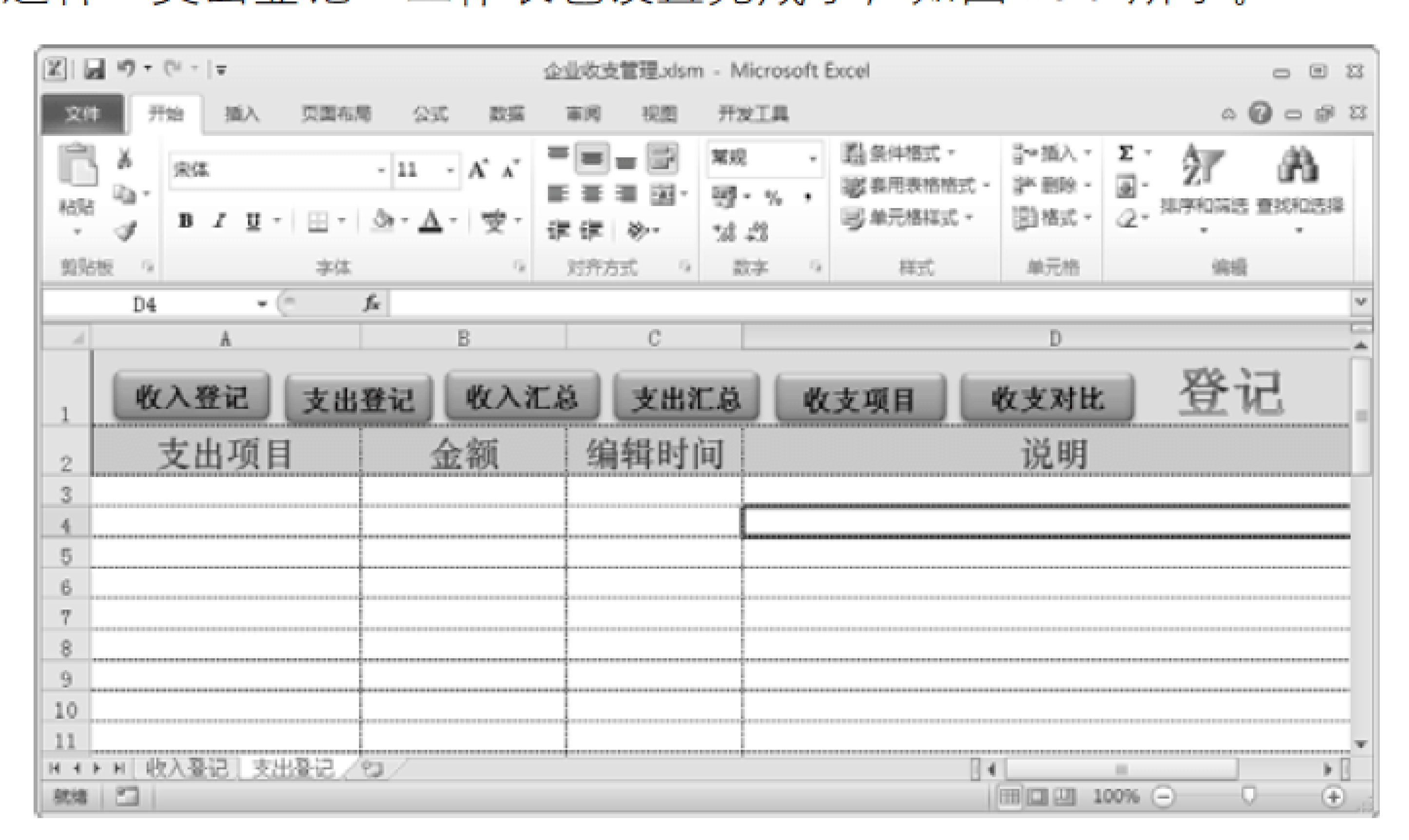


图 18-9 复制内容到“支出登记”工作表中

18.1.2 设计收支汇总工作表

收支汇总工作表包括两个工作表，分别是“收入汇总”工作表和“支出汇总”工作表，这两个工作表的格式相同。收支汇总是将企业的收支进行月份汇总和项目汇总。

01 在“企业收支管理”工作簿中，将第三个工作表标签修改为“收入汇总”，并向表中插入如图 18-10 所示的内容。

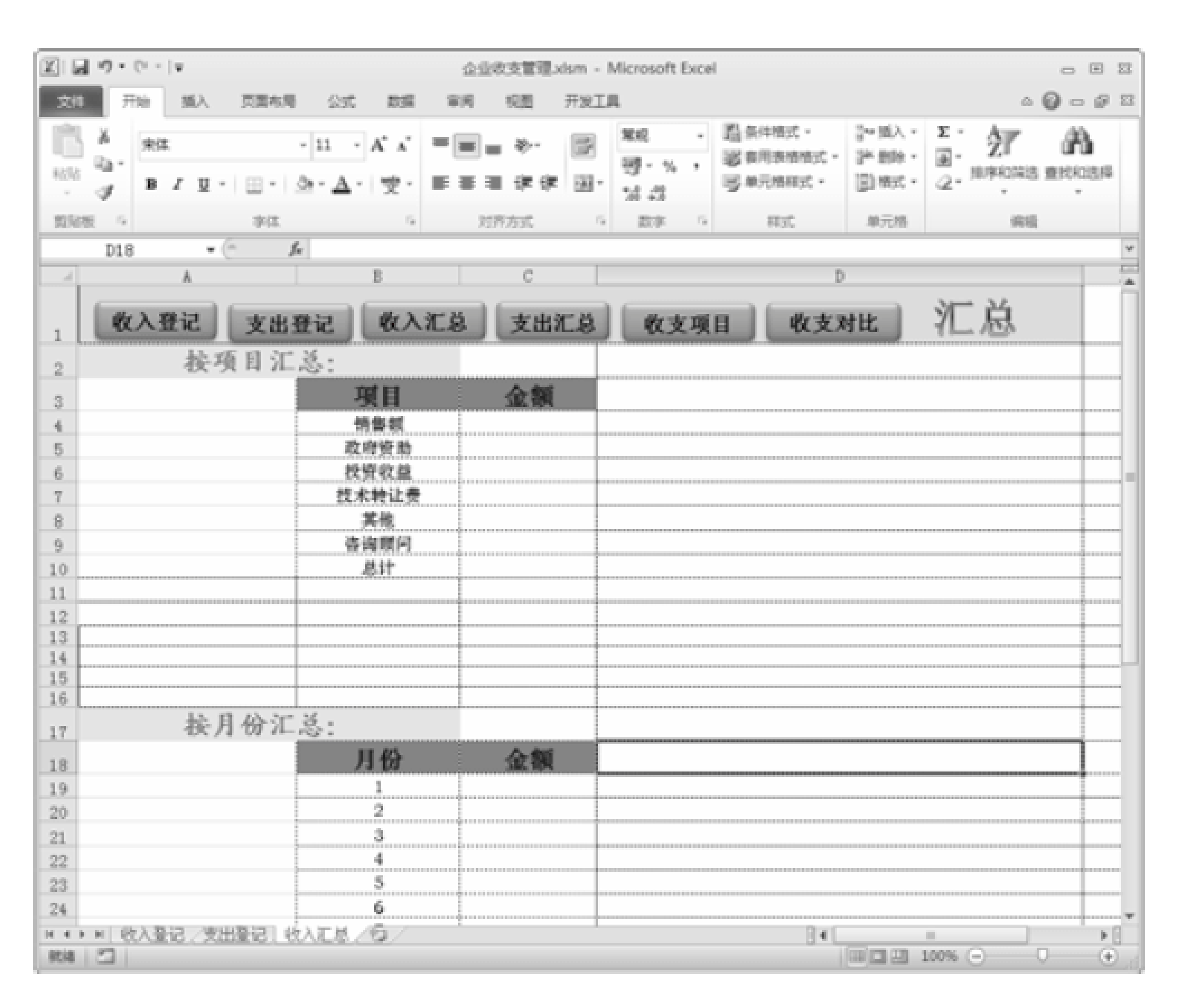


图 18-10　“收入汇总”工作表

02 选择【文件】➤【选项】菜单命令，打开【Excel 选项】对话框，切换到【高级】选项卡，在【此工作表的显示选项】组中取消选中【显示网格线】复选框，如图 18-11 所示。单击【确定】按钮，即可取消该工作表中显示网格线，如图 18-12 所示。

03 新建一个标签名称为〝支出汇总〞的工作表，然后按照前面收支登记时使用的方法，分别使用 Ctrl+A 和 Ctrl+C 的组合键对复制 〝收入汇总〞工作表中的内容，然后切换到〝支出汇总〞工作表，按下 Ctrl+V 组合键粘贴复制的内容，如图 18-13 所示。

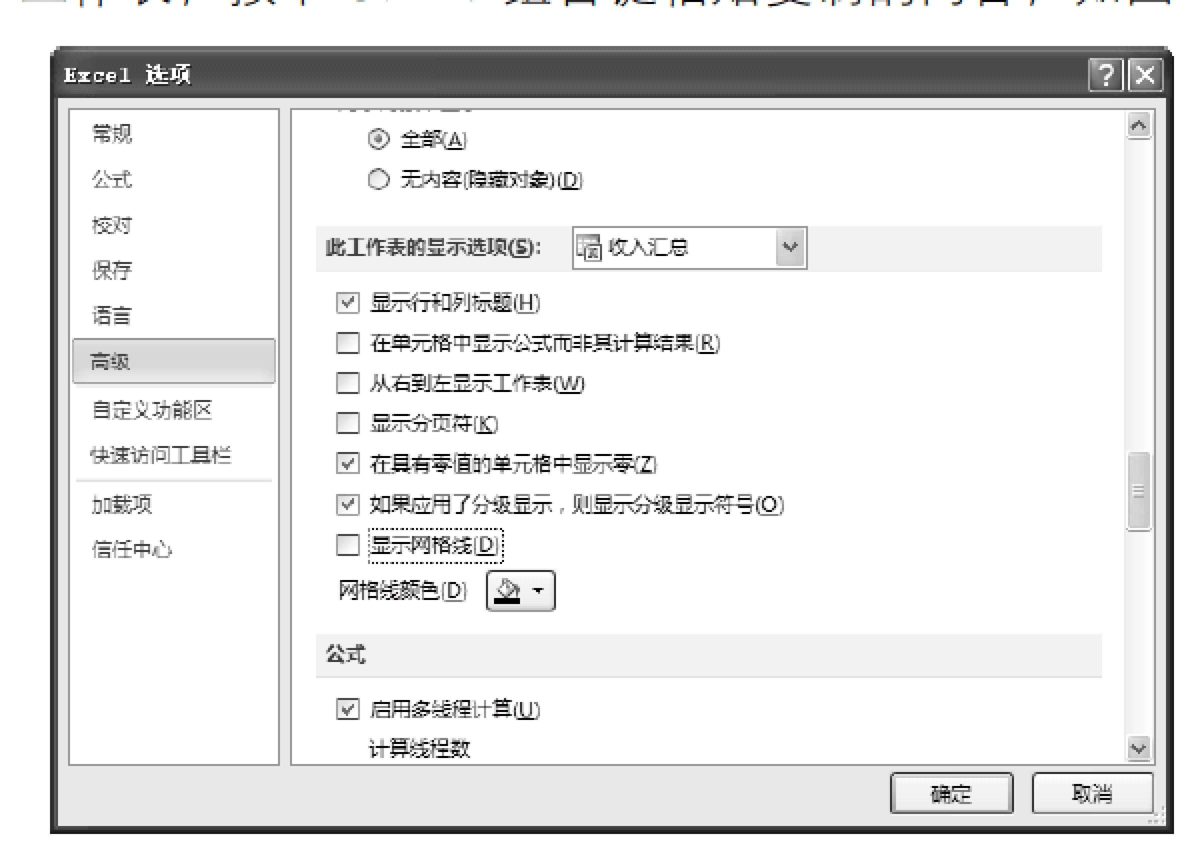


图 18-11　【Excel 选项】对话框



图 18-12　取消显示网格线

04 最后在【Excel 选项】工作表中取消选中【显示网格线】复选项，取消网格线的显示，效果如图 18-14 所示。

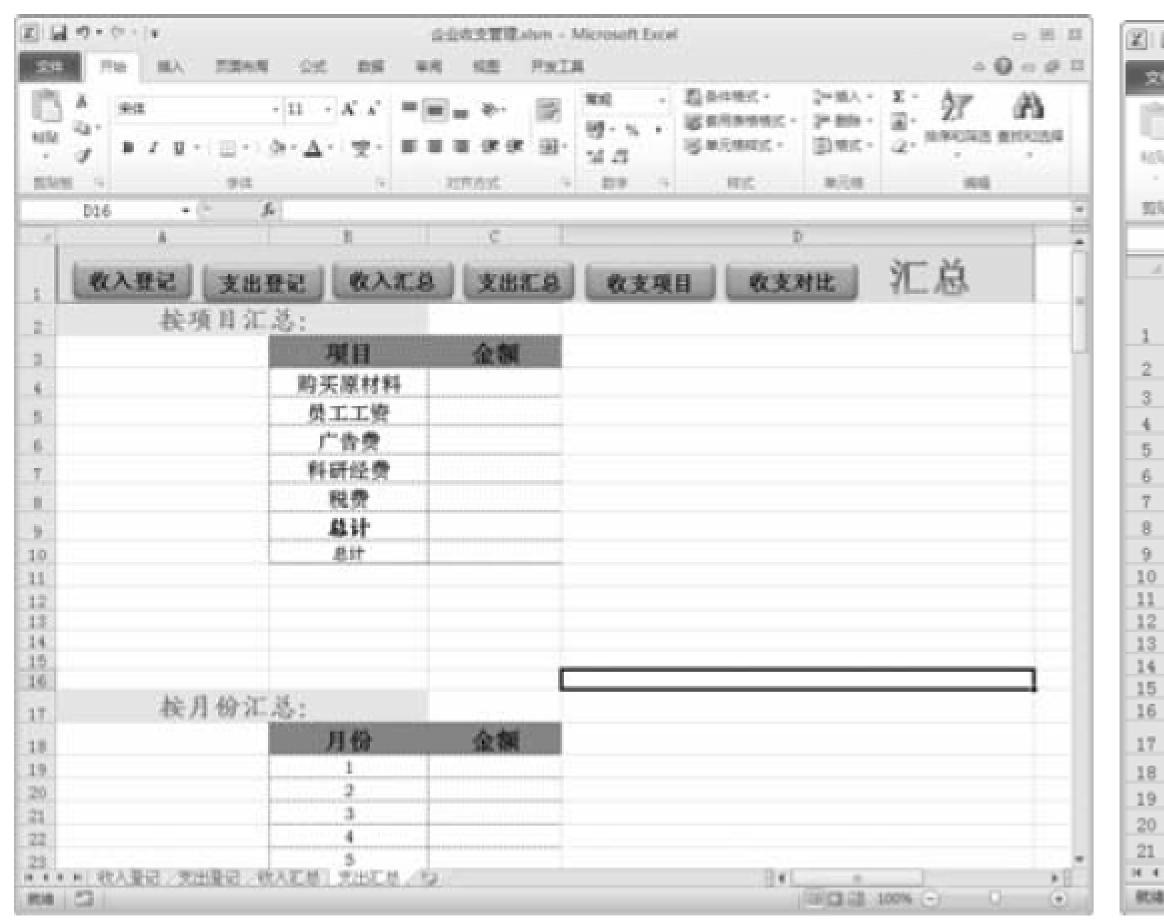


图 18-13 “支出汇总”工作表

图 18-14 取消显示网格线

18.1.3 设计“收支对比”、“收支项目”工作表

收支对比主要是比较每个月汇总的收入和支出金额，设计的“收支对比”工作表如图 18-15 所示。由于收入项目与支出项目一般不同，“收支项目”工作表就是将企业收入与支出的项目在此表中列出，以方便其他表的引用。设计的“收支项目”工作表如图 18-16 所示。

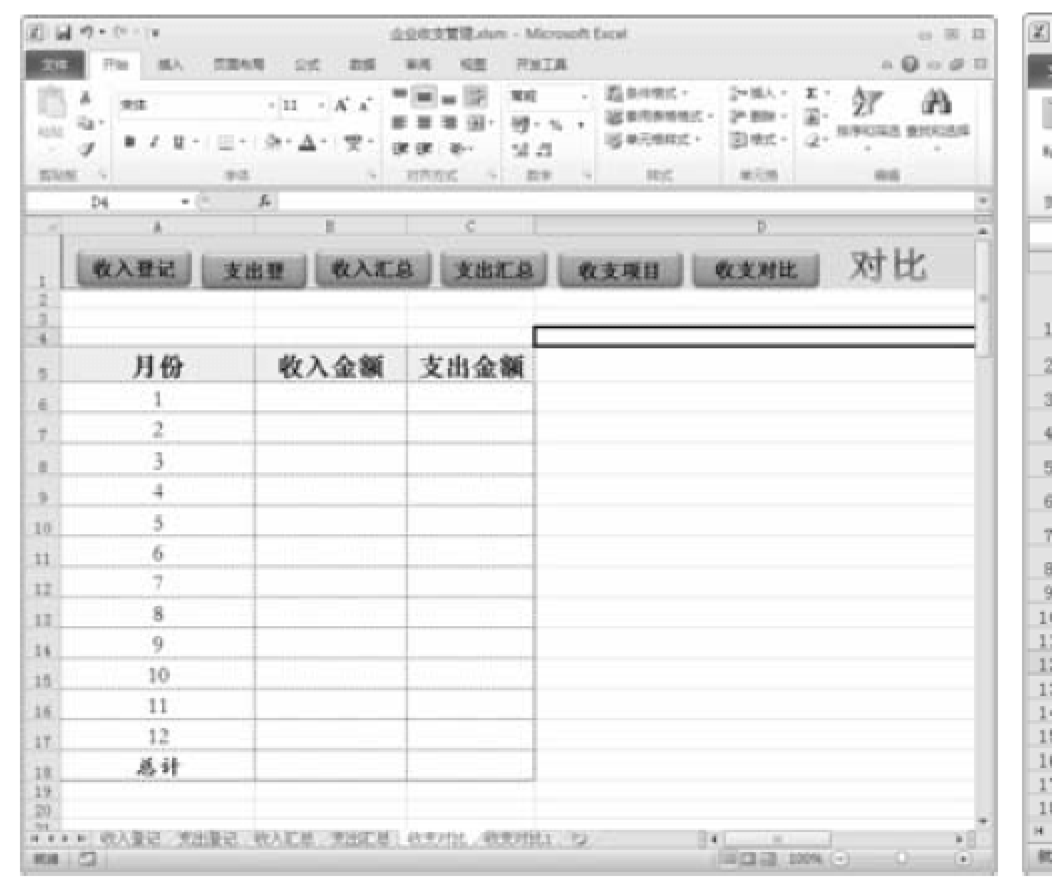


图 18-15 收支对比工作表

图 18-16 收支项目工作表



18.2 实现企业收支管理功能

在前面章节中完成了“企业收支管理”工作簿的设计，但是到现在为止，工作簿中还没

有任何内容，接下来将介绍如何在这些工作表中进行收入与支出的登记、收入与支出的汇总以及收入与支出的对比。

18.2.1　实现收支快速登记

收支登记的内容主要有收支项目与登记日期、时间以及说明文字。下面介绍以列表的形式动态显示收支项目，并实现登记日期和时间的自动添加，具体的操作步骤如下。

01 首先切换到“收支项目”工作表中，这里添加一些收支项目，如图 18-17 所示。

02 切换到“收入登记”工作表中，在【开发工具】选项卡，单击【控件】选项组中的【插入】按钮，单击列表中【列表框（ActiveX 控件)】按钮，在工作表中添加一个用于显示收入项目的列表框，如图 18-18 所示。

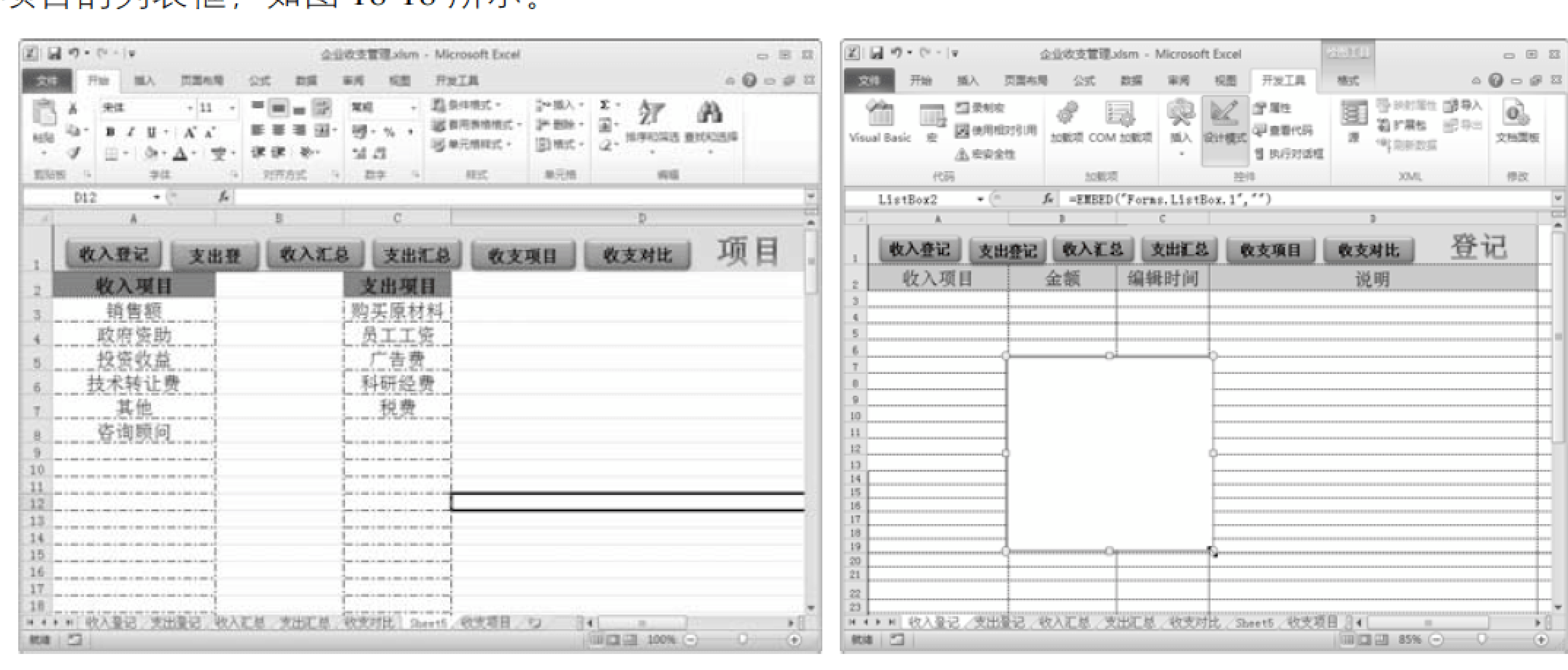


图 18-17　添加收支项目　　　　图 18-18　添加一个列表框

按 Alt+F11 组合键，打开 VBA 代码编辑窗口。在【工程资源管理器】窗口中，双击“Sheet（收入登记)”选项，打开 Sheet1 的代码编辑窗口，输入代码如下：

```
Private Sub Worksheet SelectionChange(ByVal Target As Range)
    '当前活动单元格属于项目列时自动显示项目列表框
    Static list_show As Boolean  '定义一个静态布尔变量
        If Target.Column = 1 And Target.row > 2 Then
            Call list_start      '显示列表框
            list_show = True
        Else
            If list_show = True Then Call list_end  '隐藏列表框
            list_show = False
        End If
End Sub
```

当工作表中的当前选择区域发生变化时，就会执行上述代码。该段代码表示当前活动单元格是否是第 1 列第 2 行以下的内容，如果是，就执行 list_start 过程，即显示列表框，并将静态布尔变量 list_show 赋值为 True；如果不是，则通过判断 list_show 的真假，来决定是否

调用 list_end 过程，即隐藏列表框。

在上述代码中之所以将布尔变量 list_show 定义为静态变量，是因为在过程中所用到的静态变量的值可以保留下来，在下一次调用此过程时，变量的初值是上次调用结束时被保留的值，不会被重新初始化。

03 上述代码中所用到的 list_start 过程和 list_end 过程是两个用户自定义过程，在 Sheet1 的代码编辑窗口中添加实现这两个过程的代码如下。

```
'调用收入项目列表框供选择
Sub list_start()
    Application.ScreenUpdating = False
    With ListBox1   '初始化项目列表框
        .Visible = True
        .Clear
        .Width = ActiveCell.Width
        .Height = ActiveCell.Height * 8
        .Left = ActiveCell.Left + ActiveCell.Width
        .Top = ActiveCell.Top
        .BackColor = RGB(0, 100, 252)  '蓝色
        .AutoLoad = True
    End With
    '设置项目列表框内容
    Dim row_ As Integer, Item As Range
        row  = Sheets("收支项目").[B1].CurrentRegion.Rows.count
            For Each Item In Sheets("收支项目").Range("C3:C" & row )
                ListBox1.AddItem Item.Value
            Next Item
End Sub
'隐藏项目列表框并复位
Sub list_end()
    Application.ScreenUpdating = False
    With ListBox1
        .Visible = False
        .Clear
        .Top = 1
        .Height = 1
    End With
    Application.ScreenUpdating = True
End Sub
```

04 列表框中列出了所有的收入项目，当用户双击列表框中的内容时便可以在活动单元格中插入一项收入内容，所以需要处理列表的双击事件，在 Sheet1 代码编辑窗口中添加如下代码。

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    '双击项目清单中的列表项，将其赋值于活动单元格
    Cancel = True
    ActiveCell.Value = ListBox1.Value  '将项目赋值给活动单元格
    Call list_end  '调用 list_end 过程，隐藏列表框
End Sub
```

这里参数 Cancel 在双击列表框事件发生时为 False。由于在上述过程中将此参数设为 True，所以该过程执行结束之后不再执行双击事件。

05 在 VBA 代码窗口的工具栏中再次单击【设计模式】按钮，然后关闭 VBA 代码窗口返回“收入登记”工作表，从中选择第 1 列第 2 行以下的单元格，例如选择 A7 单元格，即可得到如图 18-19 所示的效果。

在弹出的列表框中双击“销售额”选项，即可将其添加到活动单元格中，如图 18-20 所示。



图 18-19　弹出的列表框



图 18-20　双击某选项将其添加到单元格中

06 自动在“编辑时间”列插入收入项目的插入时间，在 Sheet1 的代码编辑窗口中添加如下代码。

```
Private Sub Worksheet_Change(ByVal Target As Range)
    '自动登记编辑时间
    If Target.Column = 2 And Target.Row > 2 Then
       If Target.Offset(0, -1) <> "" Then
           Target.Offset(0, 1).Value = Now()
       Else '未填写项目时弹出提示信息对话框
           MsgBox "收入项目尚未填写！", 64, "提示"
           Target.Select
       End If
    End If
End Sub
```

07 关闭 VBA 代码窗口返回到“收入登记”工作表中，例如在 B7 单元格中输入金额，选择 C7 单元格，即可得到编辑时间，如图 18-21 所示。如果没有选择收入项目，而直接输入金额，系统则会弹出如图 18-22 所示的提示信息，提示用户“收入项目尚未填写！”。





图 18-21　显示编辑时间　　　　图 18-22　系统提示信息

支出登记与收入登记基本相同，不同之处就是列表框中显示的项目来源于“收支项目”工作表中的支出列。

08 按照同样的方法，在〝支出登记〞工作表中添加一个用于插入支出项目的列表框。切换到〝支出登记〞工作表，在【开发工具】选项卡，单击【控件】选项组中的【插入】按钮，单击列表中【列表框（ActiveX 控件）】按钮 ，在工作表中添加一个名为 ListBox1 的列表框，如图 18-23 所示。

按 Alt+F11 组合键，进入 VBA 代码窗口，双击【工程资源管理器】窗口中的“Sheet2（支出登记）”选项，打开 Sheet2 的代码编辑窗口，输入如下代码：

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    '双击项目清单中的列表项，将其赋值给活动单元格
    Cancel = True
    ActiveCell.Value = ListBox1.Value  '将项目赋值给活动单元格
    Call list_end  '调用 list_end 过程，隐藏列表框
End Sub


Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    '当前活动单元格属于项目列时自动显示项目列表框
    Static list_show As Boolean  '定义一个静态布尔变量
        If Target.Column = 1 And Target.Row > 2 Then
            Call list_start  '显示列表框
            list_show = True
        Else
            If list_show = True Then Call list_end  '隐藏列表框
            list_show = False
        End If
End Sub
```

```
 '调用支出项目列表框供选择
Sub list_start()
    Application.ScreenUpdating = False
    With ListBox1   '初始化项目列表框
        .Visible = True
        .Clear
        .Top = ActiveCell.Top
        .Left = ActiveCell.Left + ActiveCell.Width
        .Width = ActiveCell.Width
        .Height = ActiveCell.Height * 8
        .BackColor = RGB(0, 100, 252)  '蓝色
        .AutoLoad = True
    End With
    '设置项目列表框内容
    Dim row_ As Integer, Item As Range
        row_ = Sheets("收支项目").[D1].CurrentRegion.Rows.count
            For Each Item In Sheets("收支项目").Range("E3:E" & row_)
                ListBox1.AddItem Item.Value
            Next Item
End Sub

'隐藏项目列表框并复位
Sub list_end()
    Application.ScreenUpdating = False
    With ListBox1
        .Visible = False
        .Clear
        .Top = 1
        .Height = 1
    End With
    Application.ScreenUpdating = True
End Sub

Private Sub Worksheet_Change(ByVal Target As Range)
    '自动登记编辑时间
    If Target.Column = 2 And Target.Row > 2 Then
       If Target.Offset(0, -1) <> "" Then
            Target.Offset(0, 1).Value = Now()
        Else '未填写项目时弹出提示信息对话框
            MsgBox "收入项目尚未填写！", 64, "提示"
            Target.Select
        End If
    End If
End Sub
```

09 关闭 VBA 代码窗口返回〝支出登记〞工作表，在 Excel 工作簿窗口中再次单击工具栏



上的【设计模式】按钮，然后任意选择第 1 列第 2 行以下的单元格，例如选择 A3，即可得到如图 18-24 所示的效果。



图 18-23　添加一个列表框控件　　　　图 18-24　显示列表框

18.2.2 实现收支自动汇总

下面将看到如何实现收支按月份和项目进行汇总，并且绘制出相应的项目比例图和月份走势图，具体的操作步骤如下。

01 在“收入登记”工作表和“支出登记”工作表中输入一些收支项目，为了便于后面的月份汇总操作，此处手动修改编辑时间，使登记的内容分布于几个月份，如图 18-25 和图 18-26 所示。

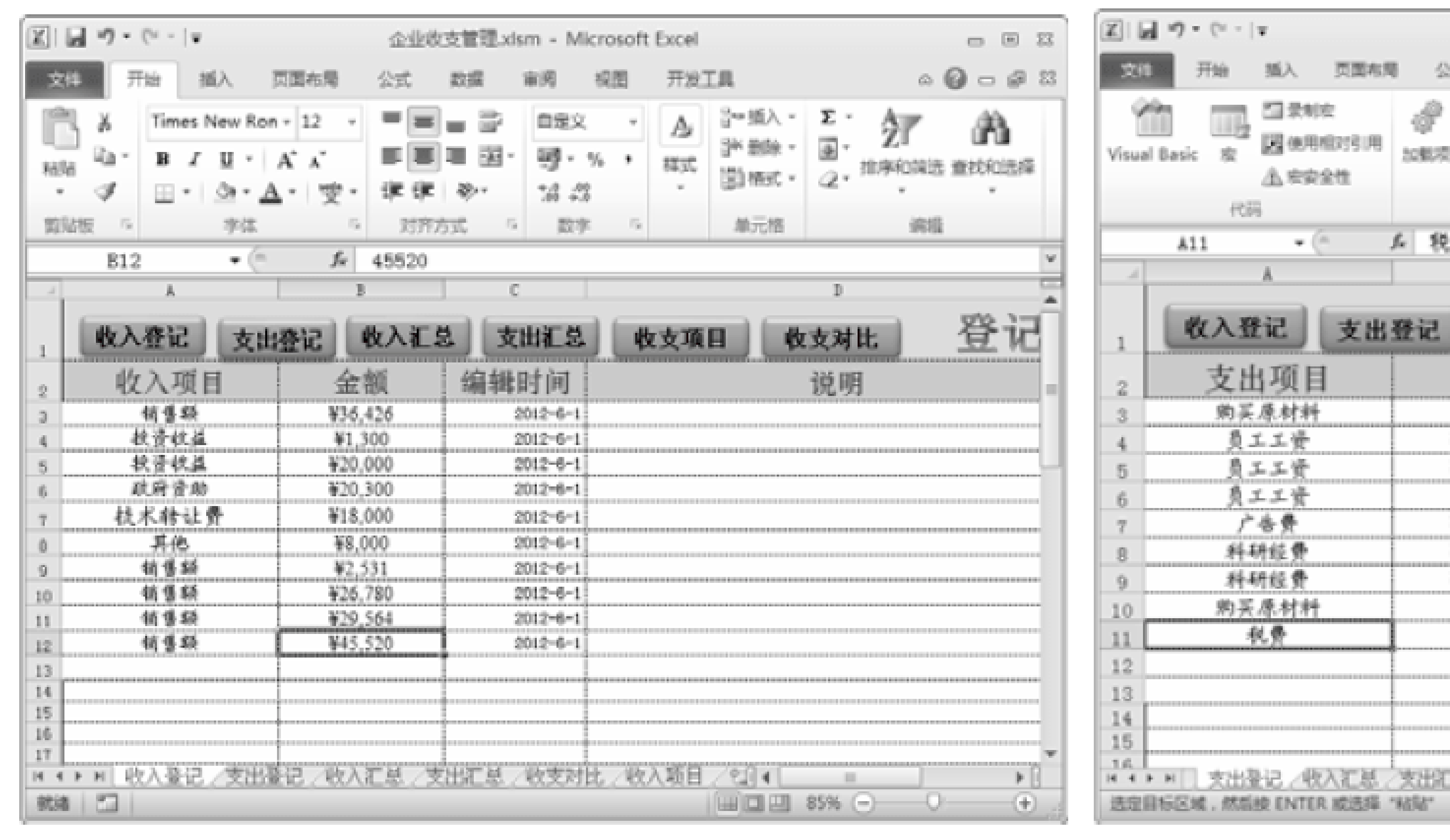




图 18-25　在“收入登记”工作表中输入收入项目　　图 18-26　在“支出登记”工作表中输入支出项目

02 按 Alt+F11 组合键打开 VBA 代码编辑窗口，从中选择【插入】➤【模块】菜单命令，插入 “模块 1”，在该模块代码窗口中添加用于汇总月收入的自定义函数 CollectByMonth1：

```
Function CollectByMonth1() '按月份汇总
    Dim count As Integer
```

```
    Dim row As Integer
    Dim month_ As Integer
    Dim Month_In(1 To 12) As Long '月收入
    For row = 1 To 12
       Month_In(row) = 0  '初始化月收入数组
    Next row
    count = Sheets("收入登记").[A1].CurrentRegion.Rows.count
    For row = 3 To count
        month_ = Month(Sheets("收入登记").Cells(row, 3))  '记录登记的月份
        Select Case month_
            Case 1
               Month_In(1) = Month_In(1) + Sheets("收入登记").Cells(row, 2)
            Case 2
               Month_In(2) = Month_In(2) + Sheets("收入登记").Cells(row, 2)
            Case 3
               Month_In(3) = Month_In(3) + Sheets("收入登记").Cells(row, 2)
            Case 4
               Month_In(4) = Month_In(4) + Sheets("收入登记").Cells(row, 2)
            Case 5
               Month_In(5) = Month_In(5) + Sheets("收入登记").Cells(row, 2)
            Case 6
               Month_In(6) = Month_In(6) + Sheets("收入登记").Cells(row, 2)
            Case 7
               Month_In(7) = Month_In(7) + Sheets("收入登记").Cells(row, 2)
            Case 8
               Month_In(8) = Month_In(8) + Sheets("收入登记").Cells(row, 2)
            Case 9
               Month_In(9) = Month_In(9) + Sheets("收入登记").Cells(row, 2)
            Case 10
               Month_In(10) = Month_In(10) + Sheets("收入登记").Cells(row, 2)
            Case 11
               Month_In(11) = Month_In(11) + Sheets("收入登记").Cells(row, 2)
            Case 12
               Month_In(12) = Month_In(12) + Sheets("收入登记").Cells(row, 2)
         End Select
    Next row
    CollectByMonth1 = Month_In  '返回月收入数组
End Function
```

03 双击【工程资源管理器】窗口中的“Sheet3（收入汇总）”选项，打开工作表 Sheet3 的代码编辑窗口，从中输入如下代码程序：

```
Private Sub Worksheet_Activate()
    Dim count1 As Integer, count2 As Integer
    Dim row As Integer, j As Integer
    count1 = Sheets("收支项目").[C1].CurrentRegion.Rows.count
    count2 = Sheets("收入登记").[A1].CurrentRegion.Rows.count
```




```
    '按项目汇总
    Dim account As Long   '总收入
    Dim Item_num As Integer  '定义收入项目数
    Dim Item(100) As String  '存储收入项目
    Dim Item_In(100) As Long  '项目收入
    account = 0
    Item_num = count1 - 2    '获得收入项目数
    For row = 3 To count1
       Item(row - 2) = Sheets("收支项目").Cells(row, 3)
    Next row
    For row = 1 To Item_num  '初始化项目收入数组
       Item_In(row) = 0
    Next row
    For row = 3 To count2
       For j = 1 To Item_num
          If Sheets("收入登记").Cells(row, 1) = Item(j) Then
             Item_In(j) = Item_In(j) + Sheets("收入登记").Cells(row, 2)
          End If
       Next j
    Next row
    For row = 4 To Item_num + 3 '将各项目收入赋值给工作表中相应的单元格
       Cells(row, 2) = Item(row - 3)
       Cells(row, 3) = Item_In(row - 3)
       account = account + Item_In(row - 3)
    Next row
    Cells(row, 2) = "总计"
    Cells(row, 3) = account  '收入总额


    '按月份汇总
    Dim month_ As Integer
    Dim Month_In() As Long
    Month_In() = CollectByMonth1
    For row = 19 To 30      '将各月份收入赋值给工作表中相应的单元格
        Cells(row, 3) = Month_In(row - 18)
    Next row
    Cells(31, 3) = "=Sum(C19:C30)"  '收入总额

    Update_Chart1 (Item_num)

End Sub
```

04 关闭 VBA 代码窗口返回到"收入汇总"工作表，就会得到按项目和月份汇总的结果，如图 18-27 所示。

05 在 E2 和 E17 单元格中分别输入"项目比例图"和"月份走势图"，如图 18-28 所示。



图 18-27　按月份和项目汇总的结果



图 18-28　输入“项目比例图”和“月份走势图”

插入饼图。选中每个项目的收入，即 B4:C8 单元格区域，在【插入】选项卡中，在【图表】选项组的【饼图】下拉列表中选择“分离型三维饼图”，在工作表中插入一个图表，如图 18-29 所示。

插入柱形图。选中 B18:C30 单元格区域，在【插入】选项卡中，在【图表】选项组的【柱形图】下拉列表中选择“簇状柱形图”，在工作表中插入一个图表，如图 18-30 所示。

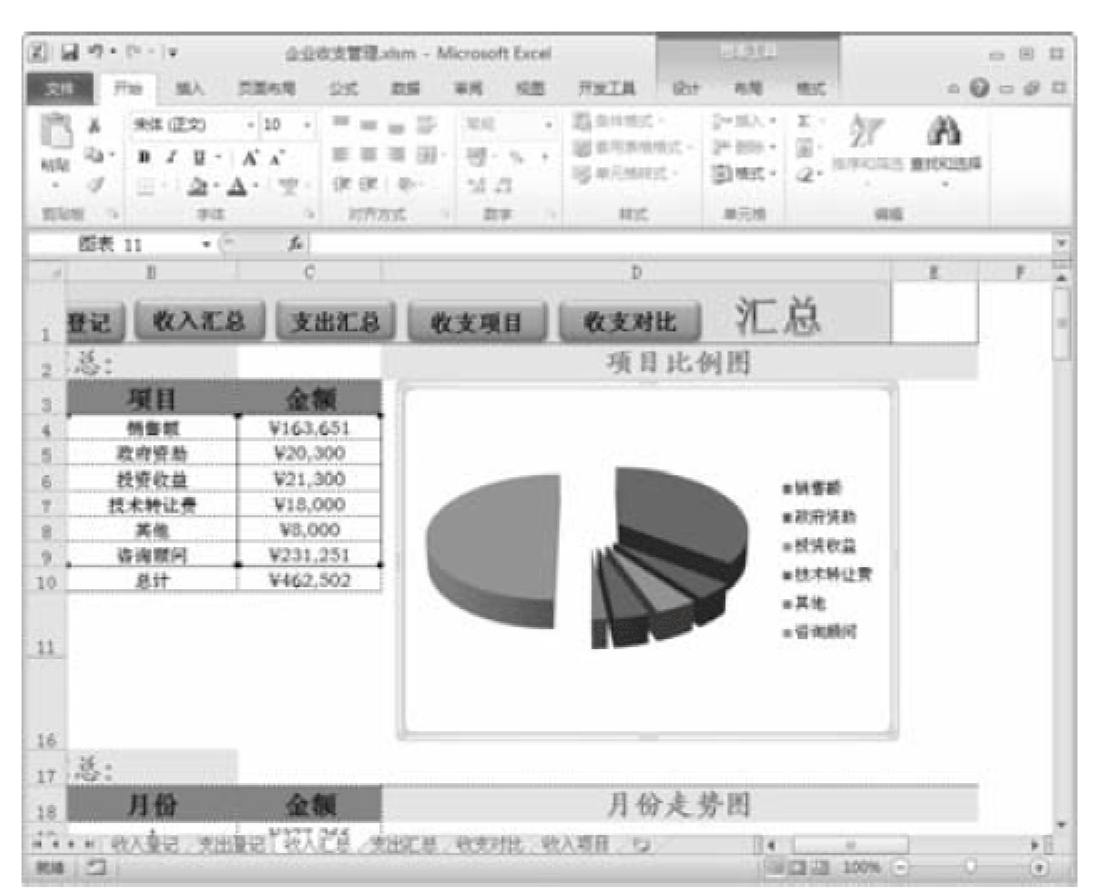


图 18-29　插入一个饼形图

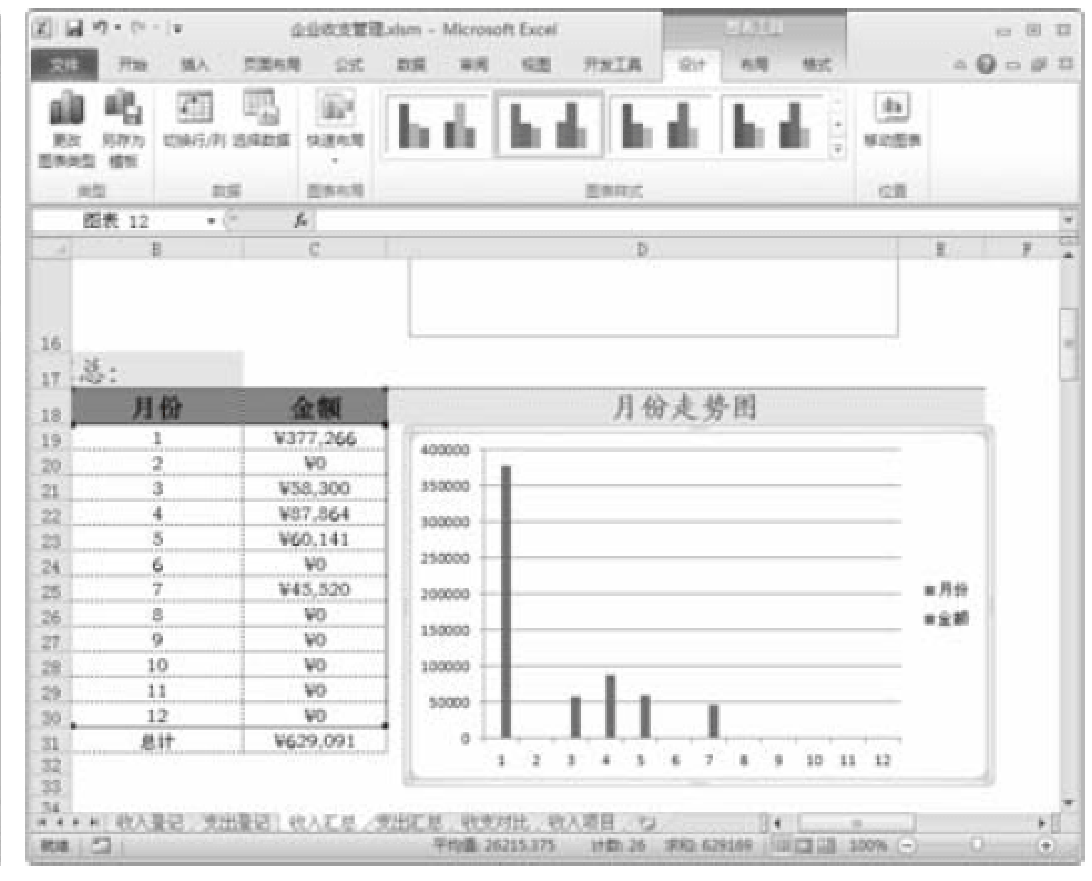


图 18-30　插入一个柱状图

06 由于年的月份就是 12 个月，不会变化，所以月份走势图的数据源就是单元格区域“B18:C30”；而项目列支可能会增加新的项目，所以项目比例图的数据源不一定就是单元格区域“B4:C8”。这就需要在程序中实现项目比例图的数据源的动态变化，在工作表 Sheet3 的代码编辑窗口中输入如下代码：

```
   Sub Update Chart1(Item num As Integer)
      Dim num As Integer
      num = Item num + 3  '最后一个项目所在行号
      Sheets("收入汇总").ChartObjects(1).Activate
      ActiveChart.SetSourceData Source:=Sheets("收入汇总").Range("B4:C" & num),
PlotBy:=xlColumns
   End Sub
```

这段代码中传递的参数 item_num 即是 Worksheet_Activate 过程中定义的项目数 item_num，为此需要在工作表 Sheet3 的 Worksheet_Activate 过程的最后调用自定义过程 Update_Chart1。

07 在工作表 Sheet3 的 Worksheet_Activate 代码框架中添加如下一行代码，用于调用 Update_Chart1 自定义过程，如图 18-31 所示。

```
Update_Chart1 (Item_num)
```

返回“收支项目”工作表，添加一个新的收入项目“咨询顾问”，如图 18-32 所示。

08 在“收入登记”工作表中添加修改“咨询顾问”项目的收入登记值，如图 18-33 所示。

09 切换到“收入汇总”工作表中，会发现项目比例图自动发生了变化，如图 18-34 所示。

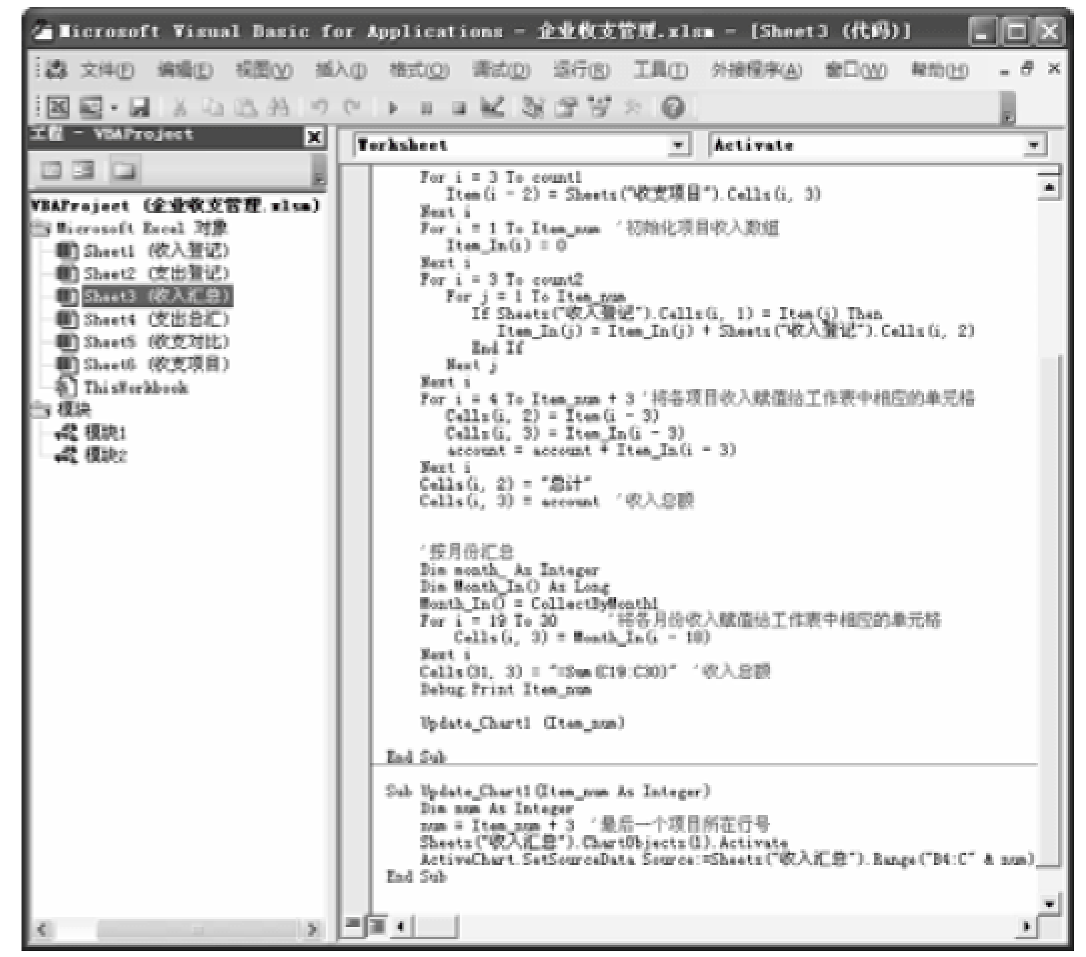


图 18-31　添加调用代码

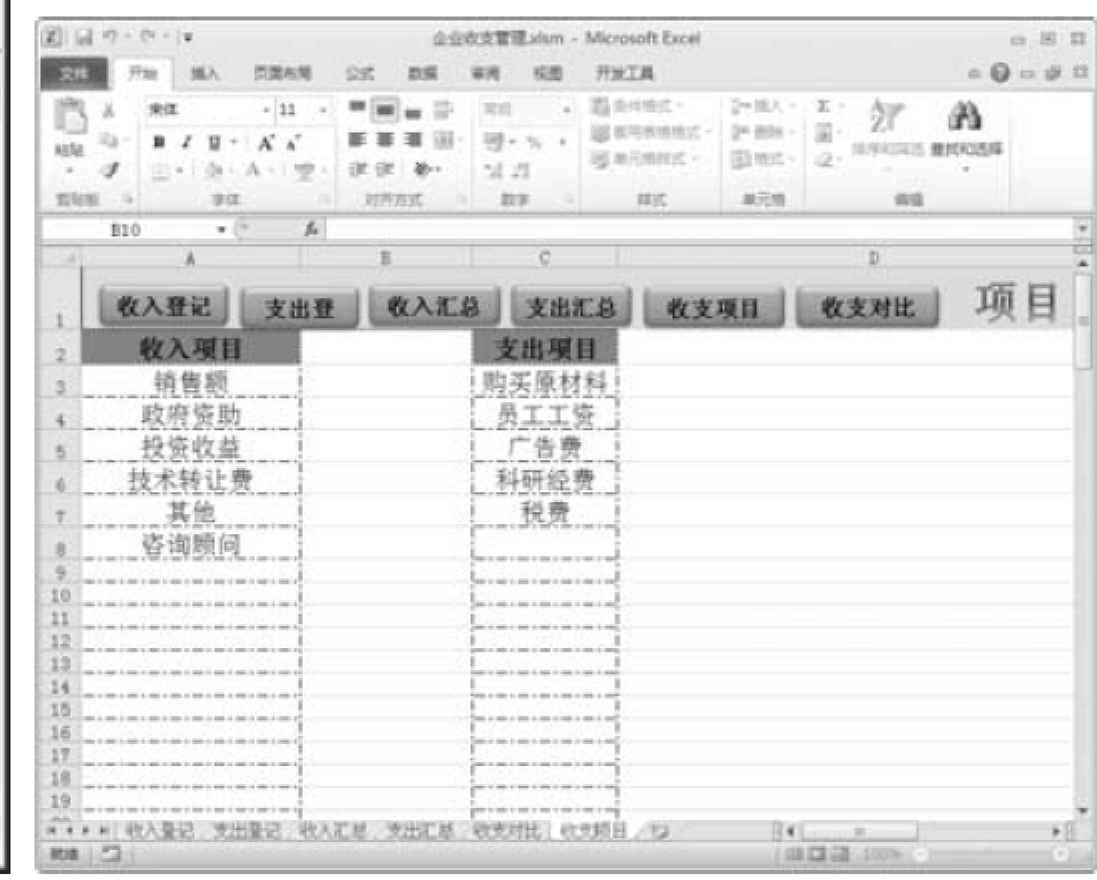


图 18-32　添加新的收入项目



图 18-33　添加关于“顾问咨询”的项目收入登记



图 18-34　项目比例图发生了变化

10 进行支出汇总。按 Alt+F11 组合键打开 VBA 代码窗口，在模块 1 代码编辑窗口中添加如下代码程序：

```
Function CollectByMonth2() '按月份汇总
    Dim count As Integer
    Dim row As Integer
    Dim month_ As Integer
    Dim Month_Out(1 To 12) As Long  '月支出
    For row = 1 To 12
       Month_Out(row) = 0  '初始化月份支出数组
    Next row
    count = Sheets("支出登记").[A1].CurrentRegion.Rows.count
    For row = 3 To count
        month_ = Month(Sheets("支出登记").Cells(row, 3))  '记录登记的月份
        Select Case month_
            Case 1
               Month_Out(1) = Month_Out(1) + Sheets("支出登记").Cells(row, 2)
            Case 2
               Month_Out(2) = Month_Out(2) + Sheets("支出登记").Cells(row, 2)
            Case 3
               Month_Out(3) = Month_Out(3) + Sheets("支出登记").Cells(row, 2)
            Case 4
               Month_Out(4) = Month_Out(4) + Sheets("支出登记").Cells(row, 2)
            Case 5
               Month_Out(5) = Month_Out(5) + Sheets("支出登记").Cells(row, 2)
            Case 6
               Month_Out(6) = Month_Out(6) + Sheets("支出登记").Cells(row, 2)
            Case 7
               Month_Out(7) = Month_Out(7) + Sheets("支出登记").Cells(row, 2)
            Case 8
               Month_Out(8) = Month_Out(8) + Sheets("支出登记").Cells(row, 2)
            Case 9
               Month_Out(9) = Month_Out(9) + Sheets("支出登记").Cells(row, 2)
            Case 10
               Month_Out(10) = Month_Out(10) + Sheets("支出登记").Cells(row, 2)
            Case 11
               Month_Out(11) = Month_Out(11) + Sheets("支出登记").Cells(row, 2)
            Case 12
               Month_Out(12) = Month_Out(12) + Sheets("支出登记").Cells(row, 2)
         End Select
    Next row
    CollectByMonth2 = Month_Out  '返回月支出数组
End Function
```

11 进入工作表 Sheet4 的代码编辑窗口，添加如下代码：

```
Private Sub Worksheet_Activate()
    Dim count1 As Integer, count2 As Integer
    Dim i As Integer, j As Integer
    count1 = Sheets("收支项目").[E1].CurrentRegion.Rows.count
    count2 = Sheets("支出登记").[A1].CurrentRegion.Rows.count

    '按项目汇总
    Dim account As Long        '总支出
    Dim Item_num As Integer    '定义支出项目数
    Dim Item(100) As String    '存储支出项目
    Dim Item_In(100) As Long   '项目支出
    account = 0
```

```
        Item num = count1 - 2    '获得支出项目数
        For i = 3 To count1
           Item(i - 2) = Sheets("收支项目").Cells(i, 5)
        Next i
        For i = 1 To Item_num  '初始化项目支出数组
           Item In(i) = 0
        Next i
        For i = 3 To count2
           For j = 1 To Item_num
              If Sheets("支出登记").Cells(i, 1) = Item(j) Then
                 Item_In(j) = Item_In(j) + Sheets("支出登记").Cells(i, 2)
              End If
           Next j
        Next i
        For i = 4 To Item_num + 3 '将各项目支出赋值给工作表中相应的单元格
           Cells(i, 2) = Item(i - 3)
           Cells(i, 3) = Item_In(i - 3)
           account = account + Item In(i - 3)
        Next i
        Cells(i, 2) = "总计"
        Cells(i, 3) = account  '支出总额


         '按月份汇总
        Dim month  As Integer
        Dim Month Out() As Long
        Month Out() = CollectByMonth2
        For i = 19 To 30 '将各月份支出赋值给工作表中相应的单元格
           Cells(i, 3) = Month Out(i - 18)
        Next i
        Cells(31, 3) = "=Sum(C4:C15)"  '支出总额

        Update Chart2 (Item num)

    End Sub

    Sub Update Chart2(Item num As Integer)
    Dim num As Integer
    num = Item num + 3  '最后一个项目所在行号
    Sheets("支出总汇").ChartObjects(1).Activate
    ActiveChart.SetSourceData Source:=Sheets("支出总汇").Range("B4:C" & num),
PlotBy:=xlColumns
    End Sub
```

12 返回到“支出汇总”工作表中，此时会得到按月份和项目汇总的支出金额，如图 18-35 所示。

13 按照前面“收入汇总”工作表中添加图表的方法，在“支出汇总”工作表中添加项目比例图和月份走势图，如图 18-36 所示。

14 在工作表 Sheet4 的代码编辑窗口中添加如下代码程序，使项目比例图的数据源动态化。

```
    Sub Update_Chart2(Item_num As Integer)
        Dim num As Integer
        num = Item_num + 3  '最后一个项目所在行号
```

```
    Sheets("支出总汇").ChartObjects(1).Activate
    ActiveChart.SetSourceData Source:=Sheets("支出总汇").Range("B4:C" & num),
PlotBy:=xlColumns
  End Sub
```

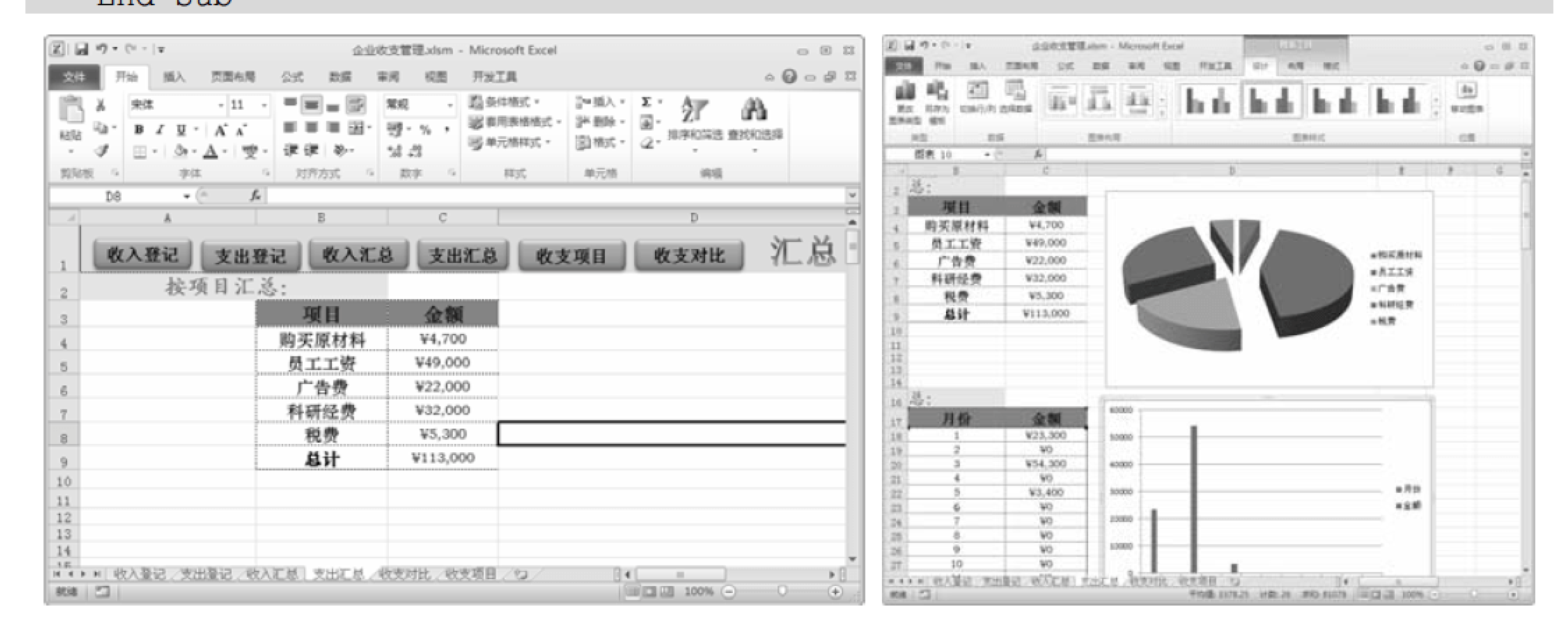


图 18-35　按项目和月份汇总的支出金额　　　图 18-36　添加项目比例图和月份走势图

15 在工作表 Sheet4 的代码编辑窗口中的 Worksheet_Activate 过程的最后添加调用 Update_Chart2 的语句，如图 18-37 所示。

```
  Update_Chart2 (Item_num)
```



```
Worksheet                                   Activate
        Item(i - 2) = Sheets("收支项目").Cells(i, 5)
    Next i
    For i = 1 To Item_num  '初始化项目支出数组
        Item_In(i) = 0
    Next i
    For i = 3 To count2
        For j = 1 To Item_num
            If Sheets("支出登记").Cells(i, 1) = Item(j) Then
                Item_In(j) = Item_In(j) + Sheets("支出登记").Cells(i, 2)
            End If
        Next j
    Next i
    For i = 4 To Item_num + 3 '将各项目支出赋值给工作表中相应的单元格
        Cells(i, 2) = Item(i - 3)
        Cells(i, 3) = Item_In(i - 3)
        account = account + Item_In(i - 3)
    Next i
    Cells(i, 2) = "总计"
    Cells(i, 3) = account  '支出总额


    '按月份汇总
    Dim month_ As Integer
    Dim Month_Out() As Long
    Month_Out() = CollectByMonth2
    For i = 19 To 30 '将各月份支出赋值给工作表中相应的单元格
        Cells(i, 3) = Month_Out(i - 18)
    Next i
    Cells(31, 3) = "=Sum(C4:C15)"  '支出总额

  Update_Chart2 (Item_num)

End Sub

Sub Update_Chart2(Item_num As Integer)
Dim num As Integer
num = Item_num + 3  '最后一个项目所在行号
Sheets("支出总汇").ChartObjects(1).Activate
ActiveChart.SetSourceData Source:=Sheets("支出汇总").Range("B4:C" & num), Pl
End Sub
```

图 18-37　添加调用 Update_Chart2 的语句



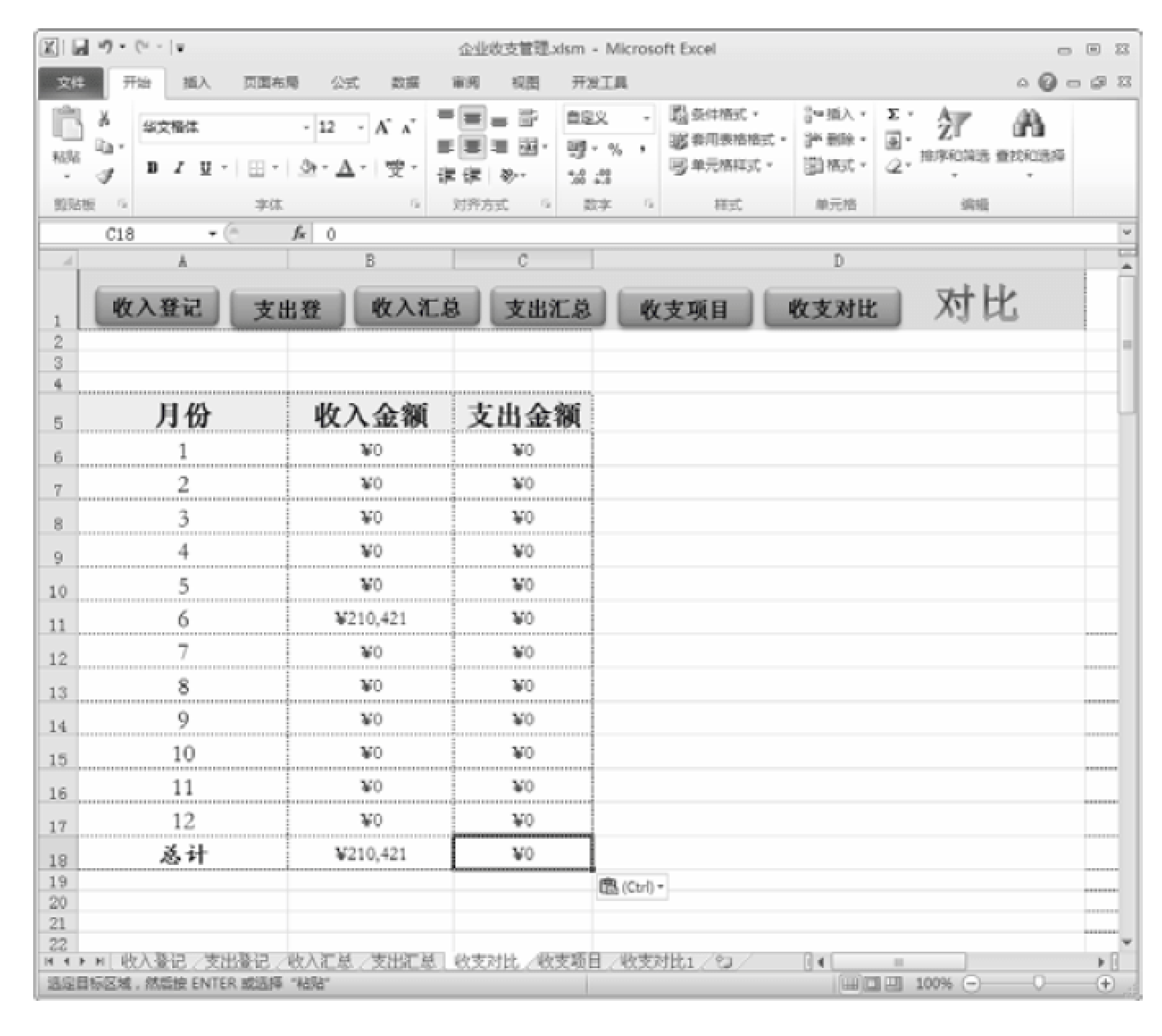
18.2.3 使用 VBA 设计收支对比图

下面介绍将收支金额按月份汇总的结果进行对比，并绘制收支对比图。

01 在 VBA 代码窗口中，打开 Sheet5 工作表的代码编辑窗口，添加如下代码：

```
Private Sub Worksheet Activate()
    Dim i As Integer
    Dim Month In() As Long
    Dim Month Out() As Long
    Dim account1 As Long
    Dim account2 As Long
    account1 = 0
    account2 = 0
    Month In() = CollectByMonth1
    Month Out() = CollectByMonth2
    For d = LBound(Month In()) To UBound(Month In())
        Debug.Print Month In(d)
    Next
    For i = 4 To 15
       Cells(i, 3) = Month In(i - 3)
       account1 = account1 + Month In(i - 3)
       Cells(i, 4) = -Month Out(i - 3)  '支出列以负值显示
       account2 = account2 - Month Out(i - 3)  '总支出也以负值显示
    Next i
    Cells(i, 3) = account1
    Cells(i, 4) = account2
End Sub
```

02 关闭 VBA 代码窗口，返回到“收支对比”工作表，即可得到按月份汇总的收支金额，如图 18-38 所示。



月份	收入金额	支出金额
1	¥0	¥0
2	¥0	¥0
3	¥0	¥0
4	¥0	¥0
5	¥0	¥0
6	¥210,421	¥0
7	¥0	¥0
8	¥0	¥0
9	¥0	¥0
10	¥0	¥0
11	¥0	¥0
12	¥0	¥0
总计	¥210,421	¥0

图 18-38 按月份汇总的收支金额

03 在 F2 单元格中输入文本“收支对比图”，并设置其字体、字号及字体颜色，然后为该单元格添加相应的边框和背景颜色，如图 18-39 所示。

04 选中 C4：D14 单元格区域，在【插入】选项卡中，在【图表】选项组的【条形图】下拉列表中选择“簇状条形图”，即可在工作表中插入一个图表，如图 18-40 所示。

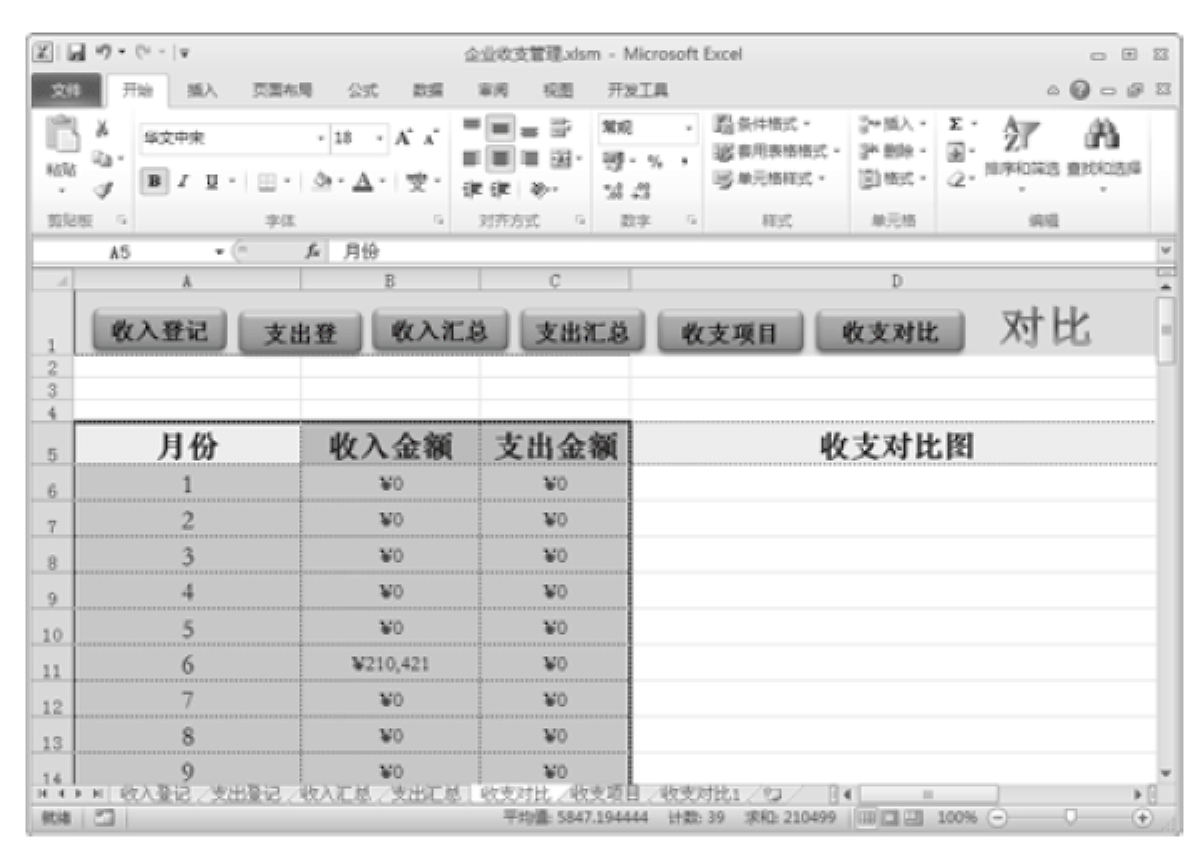


图 18-39　输入文本内容



图 18-40　插入一个簇状条形图表

18.2.4 为自定义按钮指定宏

为了便于工作表之间的切换，可以为各个工作表中添加的自定义按钮指定宏，使用这些按钮可以快速地访问工作簿中的任意工作表。

01 按 Alt+F11 组合键打开 VBA 代码窗口，从中选择【插入】➤【模块】菜单命令，插入一个名称为“模块 2”的模块，在该模块【属性】窗口中的【名称】文本框中输入“指定宏”，如图 18-41 所示，在该模块中添加如下代码。

```
Sub 收入登记()
    Sheets("收入登记").Activate
End Sub

Sub 支出登记()
    Sheets("支出登记").Activate
End Sub

Sub 收入汇总()
    Sheets("收入汇总").Activate
End Sub

Sub 支出汇总()
    Sheets("支出汇总").Activate
End Sub

Sub 收支对比()
```

```
    Sheets("收支对比").Activate
End Sub

Sub 收支项目()
    Sheets("收支项目").Activate
End Sub
```




图 18-41　修改模块名称



图 18-42　【指定宏】对话框

02 关闭 VBA 代码窗口，返回到“收入登记”工作表中，右击【收入登记】按钮，在弹出的快捷菜单中选择【指定宏】菜单命令，打开【指定宏】对话框，为其指定名称为“收入登记”的宏，如图 18-42 所示。单击【确定】按钮即可完成指定宏操作。按照这种方法，依次为其他按钮指定宏，所指定的宏的名称与按钮中所包含的文本相同。

03 切换到其他工作表中，用同样的方法为各个按钮指定相应的宏名称。

到此就完成了企业收支管理系统的设计，读者可以根据自己的需要创建一个【管理员】登录窗体，使系统的管理员通过此窗体访问、修改工作簿中的内容。